Spatiotemporal Big Data Analytics: Change Footprint Pattern Discovery

A DISSERTATION
SUBMITTED TO THE FACULTY OF THE GRADUATE SCHOOL
OF THE UNIVERSITY OF MINNESOTA
BY

Xun Zhou

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
Doctor of Philosophy

Advisor: Professor Shashi Shekhar

May, 2014

Acknowledgements

First, I would like to express my great appreciation to my advisor, Professor Shashi Shekhar, for his patient support and insightful guidance throughout my Ph.D. study here at Minnesota. He help me understand the key principles of scientific research, especially in computer science. His wisdom and experiences inspired me to confidently overcome challenges I face in both research and life.

I would like to thank all the professors who have served on my prelim oral, thesis proposal, and final defense committees: Prof. Vipin Kuamr, Prof. Mohamed Mokbel, Prof. Peter Snyder, and Prof. Snigdhansu Chatterjee. I also wish to thank all the PIs in the NSF Expeditions in Computing project: "Understanding Climate Change from Data" for their great support, guidance and helpful discussion on my work. Particularly, I wish to thank Dr. Stefan Liess and Prof. Joseph Knight for their invaluable tutorial on the climate and remote sensing background needed for my research topics. I'm also honored to have worked with the following smart and friendly colleagues in the project: Ashish Gerg, Dr. Qiang Fu, Dr. Ying Lu, Zhe Jiang, Xi Chen, Soumyadeep Chatterjee, Dr. James Faghmous, Dr. Jaya Kawale, and Varun Mithal. I'd also like to thank Prof. Jaideep Srivastava for his kind help during my Ph.D. study.

I thank all the members in the Spatial Database and Spatial Data Mining Research Group since I joined: Dr. James Kang, Dr. Pradeep Mohan, Dr. Mike Evans, Dev Oliver, Zhe Jiang, Viswanath Gunturi, KwangSoo Yang, Reem Ali, and Emre Eftelioglu, for their great help, fruitful collaborations, and warm encouragements during the past five years. I would also like to thank the alumni of our group who helped me in developing my research and academic career: Prof. Sanjay Chawla, Prof. Chang-Tien Lu, Prof. Ranga Raju Vatsavai, Prof. Yan Huang, and Prof. Hui Xiong. I also thank Kim Koffolt for valuable feedback on my papers.

In addition, I thank all the funding agencies to support my research work: National Science Foundation under Grant No. 1029711, IIS-1320580, 0940818 and IIS-1218168 as well as USDOD under Grant No. HM1582-08-1-0017, and HM0210-13-1-0005.

Also I'd like to thank all my friends at the University of Minnesota (not listed above): Jie Bao, Yanhua Li, Xiaofan Wu, and Wei Zhang for all their help in the past five years.

Finally, I would like to thank my parents for their understanding, encouragement, and support during my Ph.D. life.

Abstract

Recent years have seen the emergence of many new and valuable datasets such as global climate projection, GPS traces, and tweets. However, these Spatiotemporal Big Data (STBD) poses significant challenges for data analytics due to high data variety and candidate-pattern cardinality. One specific STBD analytics tasks is change footprint pattern discovery. Given a definition of change and a dataset about a spatiotemporal (ST) phenomenon, ST change footprint pattern discovery is the process of identifying the location and/or time of such changes in the data. This problem is of fundamental significance to a variety of applications such as understanding climate change, public safety, environmental monitoring, etc.

This thesis formally defines the spatiotemporal change footprint as a new pattern family in STBD analytics, and examined footprint patterns and related discovery techniques across disciplines via a novel taxonomy. Methods for detecting change footprints have emerged from a diverse set of research areas, ranging from time series analysis and remote sensing to spatial statistics. Existing reviews focus on discovery methods for only one or a few types of change footprints. To facilitate sharing of insights across disciplines, we conduct a multi-disciplinary review of ST change patterns and their respective discovery methods. We develop a taxonomy of possible ST change footprints and classified our review findings accordingly. This exercise allows us to identify gaps in the research that we consider ripe for exploration, most notably change pattern discovery in vector ST datasets.

To address the research gaps identified in the above analysis, this thesis further explores the computational solutions to the discovery of two specific change footprint patterns, namely, interesting sub-paths (e.g., change intervals) and persistent change windows.

Given a spatiotemporal (ST) dataset and a path in its embedding spatiotemporal framework, the goal of the interesting sub-path discovery problem is to identify all interesting sub-paths defined by an interest measure. An important application domain of sub-path discovery is understanding climate change. This thesis formally defines the computational structure of interesting sub-path discovery as a Grid-based Directed

Acyclic Graph (G-DAG). We propose a new algorithm, namely, the Row-wise Traversal (after leaf-evaluation) with Column Pruning (RTCP) which brings dramatically down the memory cost for G-DAG traversal in our earlier approaches while also reducing CPU cost. We also provide theoretical analyses of correctness, completeness and computational complexity of the RTCP algorithm. Experimental evaluation on both synthetic and real datasets show that the RTCP algorithm is always the fastest in computational time among all the proposed algorithms.

The thesis finally explores a more complicated change footprint pattern, namely, the persistent change window. Given a region comprised of locations that each have a time series, the Persistent Change Windows (PCW) discovery problem aims to find all spatial window and temporal interval pairs that exhibit persistent change of attribute values over time. PCW discovery is important for critical societal applications such as detecting desertification, deforestation, and monitoring urban sprawl. The PCW discovery problem is challenging due to the large number of candidate patterns, the lack of monotonicity, and large datasets of detailed resolution and high volume. Previous approaches in ST change footprint discovery have focused on local spatial footprints for persistent change discovery and may not guarantee completeness. In contrast, we propose a space-time window enumeration and pruning (SWEP) approach that considers zonal spatial footprints when finding persistent change patterns. We provide theoretical analysis of SWEP's correctness, completeness, and space-time complexity. We also present a case study on vegetation data that demonstrates the usefulness of the proposed approach. Experimental evaluation on synthetic data show that the SWEP approach is orders of magnitude faster than the naive approach.

The work in this thesis is the first step towards understanding the spatiotemporal change footprint discovery problem, including its formulation, computational challenges and solutions, and applications. In this thesis, we have explored automatic and efficient approaches to discovery raster-based ST change footprints, and applied our techniques on climate data in the context of understanding climate change. We conclude this thesis by exploring potential ST change patterns with new footprints (e.g., geographic feature-based footprints), alternative computational paradigms (e.g., parallel and distributed STBD analytics), their challenges and solutions, and other future research directions.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Background

Spatio-temporal Big Data (STBD), e.g., location-traces, climate observations and projection, has the potential to transform our society [2]. For example, a 2011 McKinsey report [3] estimates savings of about "$600 billion annually by 2020" from location traces of smart-phones and smart vehicles by reducing unnecessary fuel consumption in traffic congestions. Climate STBD can help prepare for the impacts of climate change by prioritizing actions to enhance climate preparedness and resilience of infrastructure. President Obama signed an executive order [4] recently to facilitate that to save major expenses such as those in the aftermath of Hurricane Sandy and Katarina.

STBD analytics techniques aim to discover non-trivial, previously unknown but potentially useful patterns and knowledge from STBD using automated and scalable computational approaches. However, STBD poses significant research challenges for spatial computing analytics and management techniques. First of all, the data-variety, data-volume, and data-variety exceed the capacity of current technologies. For example, up to 10^{13} GPS traces are collected every year from location-based services and mobile devices. It has been estimated that social network sites such as Foursquare receives approximate 3 million check-ins per day [5]. In climate science, researchers have collected data with various resolutions measuring different attributes (e.g., precipitation, temperature, vegetation cover, etc) around the Earth. Dealing with these big datasets is very challenging. Second, the candidate-pattern-cardinality exceed the capacity of

current spatial computing analytics and management technologies. For example, consider a climate dataset recoding the daily rainfall around the global at 0.5 degree by 0.5 degree spatial resolution for 100 years. The total number of records at a single location is 10^5. If we consider all the possible time intervals at this location, the number of candidate will grow to 10^{10}. For the entire dataset, the data records will increase to 10^{10}, and the number of all the three-dimensional space-time windows (i.e., rectangular area with a time interval) will be 10^{20}. In addition, the unique features of spatial and spatiotemporal data such as autocorrelation and heterogeneity violate the traditional assumptions on data models, especially the independent, identical distribution (i.i.d).

1.2 Change Footprint Pattern Discovery

One specific task in spatiotemporal big data analytics is understanding patterns related to change. This task is very important for a variety of applications, ranging from climate science to public safety. However, understanding change patterns is usually a complicated process and requires answering a number of questions, including (but not limited to):

- What is changing?
- When is it changing?
- Where is the change happening?
- Why is it changing?
- What is the impact of this change?

In this thesis, I focused on the second and third questions above, i.e., where and when changes occur. I formally define this problem as spatiotemporal change footprint pattern discovery. Given a definition of change and a dataset about a spatiotemporal (ST) phenomenon, ST change footprint pattern discovery is the process of identifying the location and/or time of such changes in the data.

The problem of spatiotemporal change footprint pattern discovery is computationally challenging. Beside the challenges of STBD analytics listed above in Section 1.1,

there are three specific computational challenges. First of all, the scales of change footprints are often unknown, and may vary significantly. For example, the global climate change may last for hundreds of years while a signal change in sensors may only take one second. Second, the change patterns may lack of monotonicity over space and time. Consider the situation where the vegetation cover in grassland decreases due to desertification. The vegetation may recover slightly due to natural variability in one or two years. Also it is possible that some sub-regions may not change as much as others. Third, it is challenging to balance interest measure effectiveness and computational scalability.

1.3 Applications

The problem of ST change footprint pattern discovery is fundamentally important for a large number of applications. We list the following potential applications of the ST change pattern discovery problem.

- In manufacturing and system monitoring, it is important to detect the time when the sensor's signal deviates from normal case, which may indicate a system failure.
- In remote sensing, a common task is to find which areas in satellite images have changed between two different times (e.g., before and after a flood). Understanding these changes may help policy maker assess the impact of disasters and respond correspondingly.
- An important mission of public health is to monitor epidemic disease outbreak. Specifically, it aims to find regions where the risk of infection increases. This can also be viewed as a change pattern discovery problem.
- Public Safety: Similar to public health, public safety applications are interested in finding the change pattern of crime risk. A region with an increase in number of crime reports may indicate the need of more policing attention.
- Ecologists and environmental scientists are interested in finding spatial change patterns of ecosystems, such as the boundaries between habitats of different species

or eco-zones, or the shrinking or expansion of certain land types, such as the desertification process.

- Climate Science: Climate science is obviously interested in a number of change related questions. The biggest one is to verify and quantify global climate change. Specifically, they may ask questions such as: when did the precipitation in the Sahel region in Africa change from normal to very limited? Is the intensity, frequency, or duration of extreme climate events (e.g., flood) changing?

This thesis mainly focused on ST change footprint patterns on the data generated in climate science applications. We also use these data to validate our proposed approaches. However, the research results may also be applied on other applications to discovery interesting change patterns.

Various research domains including time series analysis, remote sensing, and spatial statistics have investigated techniques to discovery specific types of change footprint patterns on spatiotemporal raster data. Table 1.1 shows a taxonomy of the change footprint patterns with existing techniques marked (Chapter 2). Empty cells imply the research gaps. In this thesis, we explore two patterns that had yet received much research attention before our work, namely, the change sub-paths/time series change interval pattern (Chapter 3), and the persistent change window pattern (Chapter 4) as highlighted in bold.

Table 1.1: Classification of ST Raster Change Footprints and Related Techniques

		Temporal footprint			
		Snapshot	Between few snapshots	Points in Time Series	Intervals in Time Series
Spatial footprint	Local	(N/A)	Remote sensing change detection	Change time point (e.g., CUSUM)	**Time series change interval [Chapter 3]**
	Focal	Edge Detection			
	Zonal	**Sub-path of change [Chapter 3]**			**Persistent Change Window [Chapter 4]**

1.4 Contributions of the Thesis

The main contributions of this thesis is formulating the spatiotemporal change footprint pattern discovery as a new data mining pattern family, and investigating the computational approaches to the discovery of two specific change footprint patterns within this family, namely, the interesting (change) sub-path and the spatiotemporal change windows. Specifically, we make the following contributions:

First, we proposed a novel taxonomy of change detection and discovery techniques across time series analysis, remote sensing image analysis and spatial statistics based on the footprint of the change patterns [6]. The new taxonomy classifies the spatial dimension of the footprint into local, focal, or zonal, and the temporal dimension into four categories: static, between snapshots, point in time series, or interval in time series. The thesis also examined representative change footprint discovery techniques, and analyzed gaps where research is lacking in the ST change footprint pattern family, and suggest problems that merit more exploration by the data science community.

Second, we investigated the computational approach to the discovery of interval change footprints in spatial paths/time series [7, 8]. Formally, we defined this problem as interesting sub-path discovery. We model the candidate pattern space as a novel data structure, namely, the Grid-based Directed Acyclic Graph (G-DAG), and designed an efficient graph traversal strategy called the row-wise traversal with column pruning (RTCP) on a G-DAG. Two case studies on climate data show the ability of the proposed approach to identify change phenomenon such as the footprint of ecotones and the time period of precipitation decrease in Africa. Experimental results on real and synthetics data showed that the RTCP algorithm is orders of magnitude faster than competing algorithm with significantly lower memory cost.

Finally, we addressed the problem of discovering spatiotemporal change window patterns [9], which has a zonal spatial footprint and an interval temporal footprint. We proposed a novel space-time enumeration and pruning (SWEP) algorithm which efficiently evaluate and prune candidate patterns via a two-level breadth-first enumeration. One case study was presented to where the proposed approach discovered the footprint of vegetation increase due to irrigation in Saudi Arabia during 2001-2012. Experimental results showed that the proposed SWEP algorithm significantly improved

the computational cost of straight-forward solutions.

1.5 Organization of the Thesis

The rest of the thesis is organized as follows:

- In Chapter 2 I present a novel taxonomy of spatiotemporal change footprint patterns and related techniques, with research gap analysis. This chapter also serves as a general road map for the rest of the thesis and future work after this thesis.
- Chapter 3 Presents the computational solutions to the discovery of interesting sub-path in spatiotemporal datasets.
- Chapter 4 Presents the formulation and computational solution to the persistent change window discovery problem.
- Chapter 5 Concludes the thesis and presents future research directions.

Chapter 2

Spatiotemporal Change Footprint Pattern Discovery: A Novel Taxonomy and Research Gap Analysis

2.1 Introduction

Given a definition of change and a dataset about a spatiotemporal (ST) phenomenon, ST change footprint pattern discovery is the process of identifying the location and/or time of such changes in the data. Discovering footprint patterns of change from large datasets is an increasingly important activity in application domains ranging from climate science to public health. Data science (e.g., data mining, machine learning, statistics) researchers have developed numerous techniques to facilitate the discovery of such patterns. Addressing domain specific challenges, they have often worked in distinct research settings, most notably time series analysis, image analysis, and spatial statistics. An interdisciplinary review classifying and summarizing different change footprint patterns and techniques may provide domain users valuable guidance for tool selection to solve their problems. More importantly, a survey analyzing the current research accomplishments also helps data scientists identify future research needs for data science.

Applications: Change footprint pattern discovery is an important task in a number of applications. We list a few major applications and their respective domain questions that ST change pattern discovery may help answer.

- Statistical quality control: Change pattern discovery techniques have long been applied in industrial process monitoring. The main goal is to detect system faults [10]. A typical question is: at what time did the signal change?

- Remote sensing: Remote sensing techniques provide images of an area at different times (e.g., before and after a flood). By comparing two or more snapshots of the study area, one can answer a question like "which road or bridge has been damaged?" or "which area has been flooded." This task is referred to as "change detection." [11, 12, 13] Decision makers can quickly assess the loss in a disaster and respond correspondingly.

- Public Health: An important mission of public health is to monitor epidemic disease outbreak. Specifically, it aims to find regions where the risk of infection increases [14]. This can also be viewed as a change pattern discovery problem.

- Public Safety: Similar to public health, public safety applications are interested in finding the change pattern of crime risk. A region with an increase in number of crime reports may indicate the need of more policing attention [15].

- Ecology and Environmental science: Ecologists and environmental scientists are interested in finding spatial change patterns of ecosystems, such as the boundaries between habitats of different species or eco-zones [16], or the shrinking or expansion of certain land types, such as the desertification process.

- Climate Science: Climate science is obviously interested in a number of change related questions. The biggest one is to verify and quantify global climate change. Specifically, they may ask questions such as: when did the precipitation in the Sahel region in Africa change from normal to very limited? Is the intensity, frequency, or duration of extreme climate events (e.g., flood) changing? [17]

We reviewed survey papers in various disciplines and found they mostly focused on a limited types of change footprint pattern. For example, the literature on time series

change detection [11, 18, 19, 20, 21] focuses on change(s) that occur at single time points. Remote sensing and image change detection survey papers [22, 23, 24] mainly focus on finding regions (collections of pixels) of change between two imagery snapshots. A tutorial by Wong and Neill [25] discusses the problem of "event detection" which aims to discover abnormal behavior of data. The authors covered change points in time series, spatial clusters (polygon footprint) and spatiotemporal clusters (ST volume footprint), but not changes with other interesting footprints such as temporal intervals and spatial boundaries [26, 27]. Comprehensive reviews that compare techniques across disciplines are lacking.

Classifying ST change footprint patterns and techniques across disciplines is challenging. First of all, there is no unified definition of ST change. This makes it hard to compare and contrast change discovery techniques across disciplines. Second, different research disciplines employ similar terminology to describe different phenomena. For example, the term "abrupt change in time series analysis refers to a change in the statistical distribution of data at a certain time point [18]. However, in spatial Wombling, it describes a significant difference of value across boundaries separating different spatial areas [28]. This is referred to as the synonym problem. Compounding the confusion, patterns exhibiting the same ST footprint may be named differently. For example, the output of spatial Wombling, which is a set of boundaries highlighting significant changes of value, has been called (zones of) abrupt change [28],rapid change [29], etc, by different researchers. This is referred to as the homonym problem. Finally, ST change patterns defy easy classification due to their inherently complex nature. It is also hard to select the set of key features that best classifies a particular pattern. Our Contributions: In this chapter, we define an interdisciplinary framework for ST change pattern discovery and make the following specific contributions.

- We propose a taxonomy that classifies spatiotemporal changes based on their ST footprints.
- We review representative techniques in use today by data scientists for the discovery of each change pattern.
- We analyze gaps where research is lacking in a ST change footprint pattern family, and we suggest problems that merit more exploration by the data science

community.

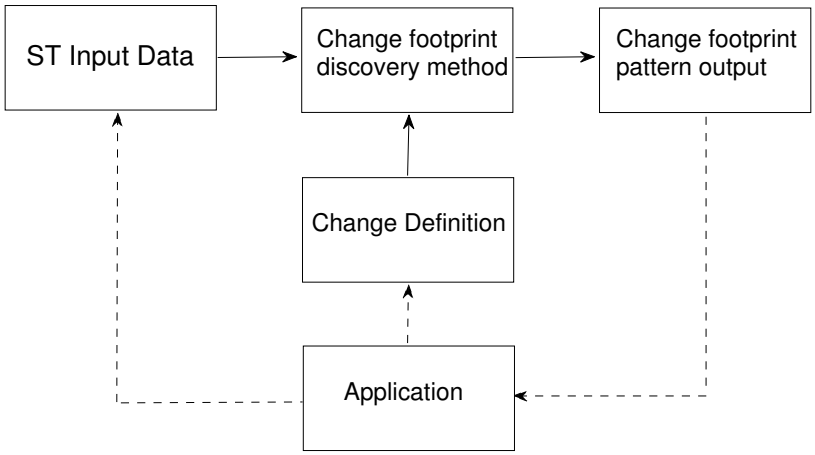


Figure 2.1: A flow chart showing ST change footprint pattern discovery process.

Scope: This chapter examines only two aspects of ST changes: where and when a change occurs. Understanding the physical mechanism of changes (why) and modeling of various change patterns from applications (how) are beyond the scope of this chapter. Also, this work focuses on the discovery of change footprint patterns from available data. Prediction methods for future changes are not addressed here. Finally, this chapter only reviews representative techniques only in the context of change footprint pattern discovery. It is not meant to provide a comprehensive survey of the large body of techniques in time series analysis (e.g., spectral decomposition), image processing, and spatial.

Organization of the chapter: Figure 2.1 shows the main components of a change footprint pattern discovery process: ST data from an application is the input of the problem. A definition of a change pattern is given based on the underlying application. Finally, a method (e.g., statistical, computational) that discovers the pattern from the data will produce the ST footprints. The rest of the chapter is organized according to

this framework: The "Data Input for Spatiotemporal Change Pattern Discovery" section surveys different spatiotemporal data types and statistical data models for change pattern discovery. The "Definitions of Spatiotemporal Change section presents four common definitions of change pattern. We propose a taxonomy of ST change footprint patterns and classify existing change footprint pattern discovery techniques in the "A Taxonomy of Spatiotemporal Change Footprint Patterns" section. In the "Future directions and research needs" section we list a few patterns that we believe merit attention by researchers in future research. Finally, we conclude the chapter with future plans.

2.2 Data Input for Spatiotemporal Change Pattern Discovery

Data generated by various applications that are used for change footprint discovery include temporal data, spatial data, and spatiotemporal data. Temporal data here refers to time series. Spatial data can be categorized into two types, namely, spatial raster data, and spatial vector data [30]. Spatiotemporal data include raster series, and spatiotemporal vector data.

A time series is a sequence of data points, measured typically at successive time instants spaced at uniform time intervals (e.g., second, day, year, etc). Many applications generate time series. For example, the readings of a sensor that monitors the quality of products form a time series. The yearly rainfall amounts in a spatial region is also a time series. Figure 2.2 shows an example of a time series generated by climate science data. It represents the annual precipitation anomaly (mean value removed from real value) from 1900 to 2010 in the Sahel region of Africa [31].

Spatial raster data represents the space as a finite grid structure (i.e., spatial framework) where a number of functions representing application specific non-spatial features are assigned to each grid. For example, the longitude-latitude system is a spatial framework; the data representing the precipitation amount over the surface of the world is a typical spatial raster dataset. A satellite image is also a typical spatial raster dataset. Figure 2.3 is an example of raster data. It represents the vegetation cover in Africa measured in normalized difference vegetation index (NDVI) value where each pixel equals approximately 8km by 8km on the ground [32, 33, 34].

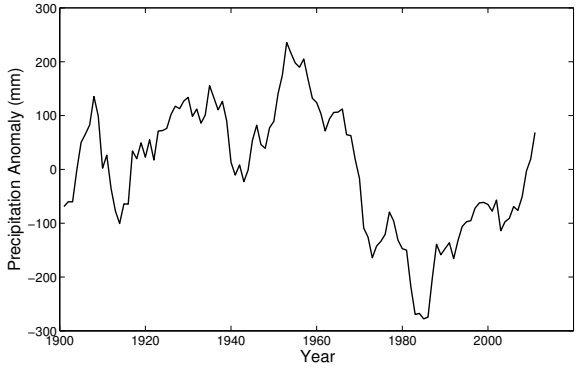


Figure 2.2: An example of time series from climate science.

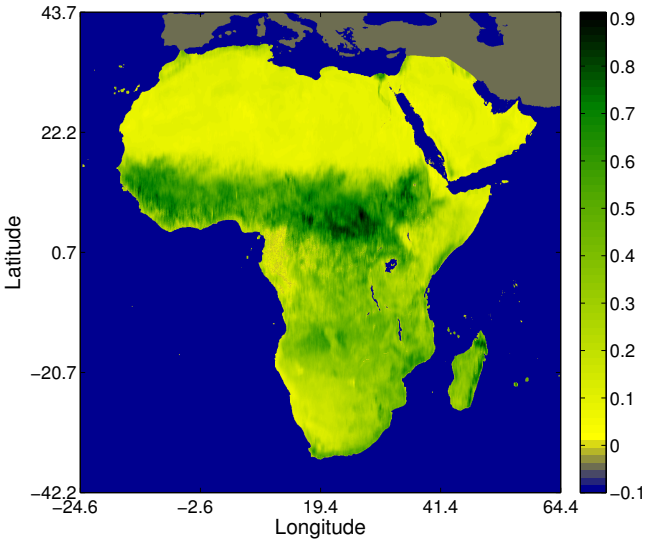


Figure 2.3: A spatial raster dataset showing vegetation cover (in NDVI value) of Africa (best viewed in color).

Relationships and interactions between different spatial raster fields are specified by field operations [35]. Depending on their scale, operations may be local, focal, or zonal [36]. Local operations determine the output at a single location depending on attributes at this location. For example, "find locations with a precipitation greater than 500mm is a local operation. Focal operations determine their output based on attributes in an assumed small neighborhood of an input location. For example, computing the gradient of precipitation value over the worlds land surface is a focal operation. Zonal operations usually employ aggregate operators of locations in a region. For example, determining the average precipitation in Africa is a zonal operation.

Spatial vector data uses geometric shapes to represent spatial objects. These shapes include points, line segments, polygons, as well as combinations of these shapes. For example, cities in a map can be represented as points. Road segments are usually represented as line segments. Counties, States, and countries are often modeled as polygons. Figure 2.7 shows an example of spatial vector data where each country in the world is represented by a polygon object and the GDP growth in 2011 of each country is an associated attribute [37].

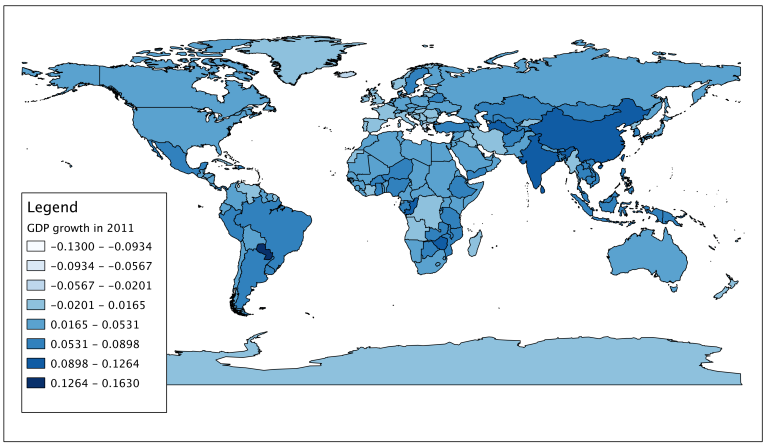


Figure 2.4: A spatial vector dataset showing GDP growth in countries around the world in 2011 (best viewed in color).

Spatiotemporal data are associated with temporal information in addition to the spatial data types. An ST raster series represents a spatial field at a number of successive

snapshots. For example, a sequence of satellite images showing the vegetation cover in Africa taken every other week forms a spatial raster series dataset. From a temporal perspective, the data at each location/pixel of the raster field also forms a time series of the same length [38]. Thus this type of dataset is also referred to as the spatial time series [39].

ST vector data are objects with both spatial and temporal information. For example, the disease reports in the state of Minnesota in the past five years form a point-collection sequence where the locations of the points may vary. The historical coverage of a birds habitat also forms a vector series, where the habitat in each year can be represented as a polygon.

Statistical Models of ST Data: Time series are traditionally modeled as independent, identically distributed (i.i.d.) samples drawn from an underlying distribution. For example, it is quite common to assume that the readings of a sensor follow a Gaussian process9 where the deviation between each reading and a fixed mean value follows a normal distribution. However, it is often the case that time-referenced attributes are auto-correlated, meaning that the value at time t is dependent on the value at time $t-1$. More complicated models of time series have been applied to model temporal data, such as the Markov chain, where the probability distribution of a value at time t depends only on the value at $t-1$ [40].

Spatial data were viewed by traditional statistics as i.i.d. data samples from a distribution. In contrast, spatial statistics, a branch of statistics that studies the modeling and analysis of spatial data, propose different models to honor the unique nature of spatial data, i.e., spatial autocorrelation and spatial heterogeneity. First of all, spatial data is highly self-correlated. This has been recognized as a fundamental observation in geography and named as the first law of geography: "Everything is related to everything, but nearby things are more related than distant things. [41] Spatial heterogeneity refers to the fact that an underlying process varies from place to place.

As a result, three spatial statistical models have been developed to model spatial data, namely, the geostatistical model, the lattice model, and the point process model [42]. The geostatistical model deals with a continuous spatial surface with discrete sampled locations (e.g., ground observational stations to record rainfall data). Tools such as Kriging are used to interpolate un-sampled values. The lattice model

(a.k.a. the areal model) deals with continuous space partitioned into regular grids or irregular polygons (e.g., the counties in Minnesota). Interactions between partitions are characterized by the spatial neighborhood relationship (e.g., topological connectivity). Data models such as the spatial autoregressive model (SAR) and Markov random field (MRF) can be applied on such datasets. Finally, point process is used to model the distribution of spatial points in a spatial framework. For example, crime locations in a city can be modeled as a point process. The distribution of points may be completely random, clustered, or de-clustered.

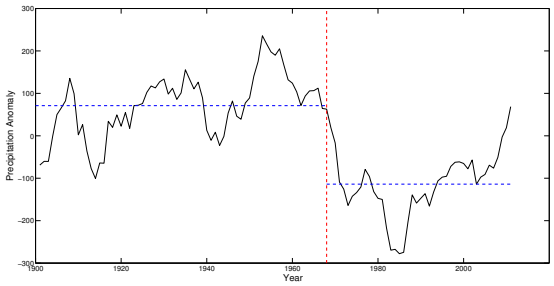
2.3 Definitions of Spatiotemporal Change

Although the same term "change is used to name patterns in various applications, the underlying phenomena may differ significantly. This section briefly summarizes four main ways for defining a change in data. Since the modeling of change is not the focus of this chapter, our review of the definitions should not be considered comprehensive.

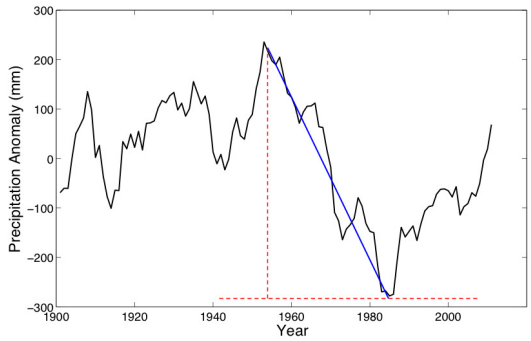
Specifically, change in data can be defined mainly in the following four different ways.

Change in Statistical Parameter: Data in some applications are assumed to be random samples drawn from an underlying process. A change is thus defined as a shift in the statistical distribution of the data. For example, in statistical quality control, sensor readings are expected to follow a certain statistical distribution (e.g., Gaussian) [18]. If a fault occurs, the mean or variance of data will change. This type of definition may make different assumptions on the underlying data distribution. Definitions with parametric models assume that the underlying distributions are of a certain kind (e.g., Gaussian distribution) while definitions with non-parametric models do not. Figure 2.5(a) shows an example of a "change of mean in a time series. As can be seen, the mean of the data before and after the highlighted time points are significantly different.

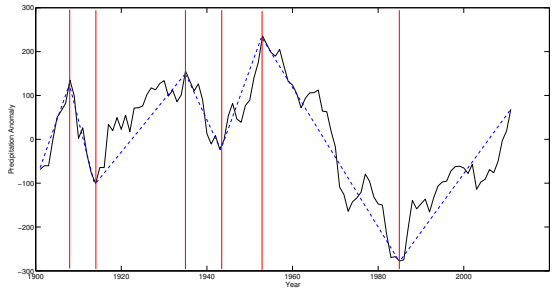
Change in actual value: The second type of definition is based on the actual values of the data. The definition of change is initially modeled (mathematically) in calculus, where a difference between a data value and its neighborhoods in location or time is viewed as a change. In a one-dimensional continuous function, the magnitude of change is usually characterized by the derivative function, while on two-dimensional surfaces,



(a)



(b)



(c)

Figure 2.5: Three sets of results for one time series dataset using three different definitions of change. (a) Statistical parameter change in a time series. (b) Value change in a time series. (c) Change in model fitted on a time series.(best viewed in color).

a change is usually characterized by the magnitude of the gradient. For a discrete function (e.g., time series), the change between two data points can be characterized by the slope of the line connecting the two. For example, in spatial statistics, boundary analysis (a.k.a. spatial Wombling) is done by finding the significant changes between neighboring locations. Figure 2.5(b) a "change of value in a time series. The time period highlighted has a steep slope, indicating it is a change.

Change in models fitted to data: A third type of definition focuses on the change in the trend/behavior of the data. A number of function models are fitted to the data where a change in one or more of the models is defined as an instance of change [43]. For example, climate scientists studying the trend of global precipitation want to be able to detect a turning point where the rainfall changes from increasing to decreasing. In such scenarios, a time series can be fitted using a certain number of straight lines by minimizing the error (e.g., least square error). Hence, a change in the data is defined as a discontinuity between two consecutive linear functions. The models can also be non-linear (e.g., polynomial) [44]. Figure 2.5(c) illustrates a "change in linear models in a time series. The highlighted time points represent "change" since they are the break points between different linear segments fitted to the data.

Change in derived attributes: Some applications define change patterns indirectly. First, they establish a classification or prediction model of the data. Then they run the model and derive new attributes, such as predicted value or a categorical class label of the data. For example, in time series change detection, a predictive model can be learnt based on a training dataset. The future values are then predicted and compared with actual values. A difference between the prediction and actual value is considered a change [45, 46].

2.4 A Taxonomy of Spatiotemporal Change Footprint Patterns

We are now ready to propose a taxonomy of the change footprint patterns from space-time perspective. We also describe representative techniques used to discover these patterns. Since the goal of this work is to identify broad gaps in the research, readers interested in a more comprehensive discussion are invited to consult more specialized

literature on specific change patterns. Also, we restrict the discussion to the context of change footprint pattern discovery. The larger realm of general techniques in time series analysis and image processing (e.g., trend analysis, spectral decomposition) is not included here.

We begin by classifying change footprints along two dimensions: temporal and spatial. Temporal footprints are of four types, namely, single snapshot (T1), set of snapshots (T2), point in a long series (T3), and interval in a long series (T4). Single snapshot (T1) means that the change is purely spatial without any temporal context. "Set of snapshots (T2) indicates that the change is between two or more versions of the same spatial field, e.g., different satellite images of the same region. T3 refers to a single time instance in a long series of data. T4 is a long time period in a long series of data.

Spatial footprints in our taxonomy are classified as raster-based or vector-based. Specifically, raster footprints are further classified based on the scale of the pattern, namely, local, focal, and zonal patterns. Vector-based patterns are further classified into four types, including point(s), line(s), Polygon(s) and network footprint patterns. In each of the two parts, we examine all the combinations of the four temporal footprints and the corresponding spatial footprints. In addition, purely temporal patterns (e.g., change points in a time series) are considered as having local spatial footprint and are discussed under the raster-based change footprints only.

2.4.1 Taxonomy of Change Patterns with Raster-based Spatial Footprint

For raster spatial data, we classify the change footprints based on whether their scale is local, focal or zonal (Table 2.1). A local change footprint (R1) involves only the attribute at individual locations. A focal change footprint (R2) is between a location and a spatial neighbourhood of it. Change patterns with a zonal footprint (R3) refer to the change that occurs in a spatial region (collection of locations as a whole pattern).

As can be observed in Table 2.1, our taxonomy yields 11 types of change patterns. Among these, we know that three groups have been well studied by the research community, namely, purely spatial change patterns (R2T1, R3T1), time series change patterns (R1T3, R1T4) and image (snapshot) change patterns (R1T2, R2T2, R3T2). In contrast,

Table 2.1: Classification of ST Raster Change Footprints with Examples of Typical Questions. Empty Cells indicate Patterns that Yet to be Studied in Depth

		Temporal footprint			
		Time point (T1)	Between few snapshots (T2)	Points in Time Series (T3)	Intervals in Time Series (T4)
Spatial footprint	Local	R1T1 (N/A)	Remote sensing image change detection (R1T2, R2T3, R3T2)	Change point detection (e.g., CUSUM) R1T3	Change interval discovery on time series R1T4
	Focal	Edge Detection (e.g., Lattice Wombling) R2T1			
	Zonal	Interesting sub-path discovery (e.g., ecotones) R3T1			

remaining patterns have received little attention in the literature of change pattern discovery. We first illustrate the known patterns and major discovery techniques and then discuss the remaining ones in the future research needs sections.

Patterns with Purely Spatial Footprint

Focal Spatial Change (R2T1): A focal spatial change pattern describes a change that occurs between a location and its spatial neighborhood. For example, given a contiguous spatial field, a focal spatial change pattern may be defined as a collection of locations with high gradient. This pattern can be used to characterize phenomena such as the boundaries between different ecological zones where environmental attributes (e.g., gene frequency) change sharply [16], or as disconnections of value between different soil types [47]. Previous investigation of such patterns has given various names to it, including "spatial boundary, "spatial barriers [48, 49], "spatial (zones) of abrupt change [50], "spatial rapid change [29], etc. A common goal of such work is to find locations with "significant difference or "abrupt change against its neighboring locations. By finding a collection of such locations, one may draw a boundary or curve between different homogeneous regions. First addressed by Womble [26], the spatial boundary analysis problem is also known as the spatial Wombling problem. In ecology study, it is also referred to as "edge detection [51].

Spatial Wombling techniques for the discovery of focal spatial change footprint have

been developed. A simple approach is called lattice Wombling [16, 51]. Given a raster spatial data field, the lattice Wombling algorithm evaluates all the grid intersections by computing the change magnitude based on the values in the four surrounding cells. In a regular gridded field, the magnitude can be computed as the mean gradient along the four directions. Connecting grid intersections with the top k% will generate boundary linear change footprint pattern. Figure 2.6 shows an example of a spatial raster field and two collections of focal changes discovered by lattice Wombling.

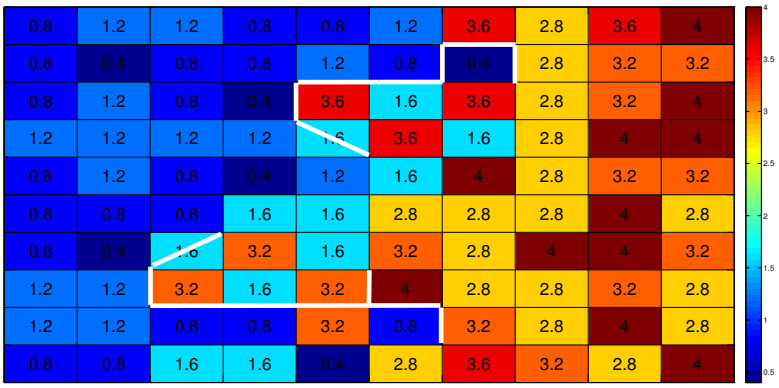
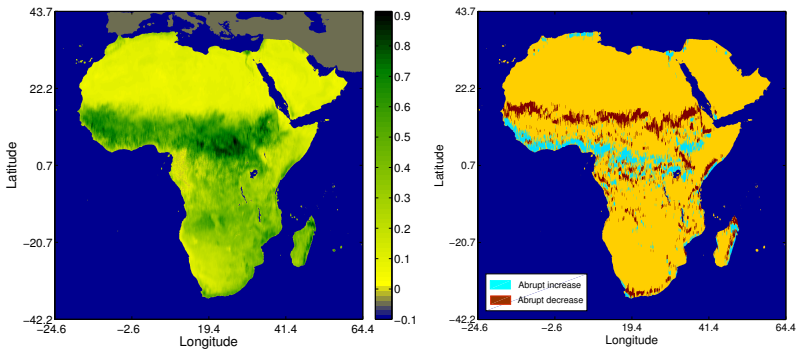


Figure 2.6: An example of lattice Wombling results on a raster field (best viewed in color).

Zonal Spatial Change (R3T1): A zonal spatial change pattern describes a spatial zone in a raster field where a transition of data attributes occurs. This pattern may characterize phenomena such as rapid environmental change across different ecological zones. For example, the Sahel region in Africa is a transitional zone between the Sahara Desert and the tropical savanna and grassland. Vegetation cover increases rapidly from north to south. These areas are also referred to as ecotones [52]. Compared to the boundary footprint formed by a collection of focal changes, a spatial zonal change footprint pattern has two-dimensional footprint and exhibits more information about the spatial process within it. This also distinguishes the zonal change footprint pattern from the boundary/line patterns discovered by edge detection or cartography line

generation [53].

Zhou et al. proposed an indirect way to approach this problem by discovering change sub-paths along orthogonal (e.g., longitudinal and latitudinal) directions [7]. The problem is converted to finding "interesting sub-paths in each spatial path. Given a spatial path, an algebraic (non-monotonic) interest measure, and a Boolean test, the technique employs a sub-path enumeration and pruning (SEP) approach to efficiently find all the dominant interesting sub-paths. It enumerates and evaluates all the sub-paths in a top-down manner with proper pruning. This computational framework allows the user to specify interest measures to define the pattern. For example, a change sub-path can be modeled by a slope or high linear regression coefficient. In the same work, the authors also proposed an algebraic interest measure called a "sameness degree based on an aggregate function of piecewise difference. Figure 2.7(a)shows the African vegetation cover dataset (a) and the longitudinal change (north to south) sub-paths outline the footprint of the Sahel region (marked in red), as shown in Figure 2.7(b).



(a) (b)

Figure 2.7: An example of spatial zonal change footprint (best viewed in color). (a) Vegetation cover (in NDVI) in Africa, August, 1981. (b) Footprints of spatial zonal change patterns with longitudinal changes in vegetation cover of Africa.

Patterns with Purely Temporal Footprint

Time point change (R1T3): A time point change refers to a change occurring at a single time point or in one unit time intervals in a time series. Time point change

patterns have been explored in various applications to find phenomenon such as the time instance of system fault using sensor signal series, year of abrupt climate change in rainfall and temperature time series, and time of land cover change in remote sensing satellite data series at a location. As discussed previously, such changes can be defined as a shift in statistical parameters (e.g., mean), a change in the model fitted to the data, or a change in derived attributes (e.g., prediction results). In the literature of time series analysis, this problem is also referred to as "change point detection", "abrupt change detection", or simply "time series change detection".

There is a large body of literature on techniques that identify time point change patterns in time series [11, 54, 55, 56, 57, 58, 59]. Several survey papers have classified and reviewed existing techniques on change point detection [18, 19, 20, 60, 61, 62, 63, 64, 65, 66, 67, 68]. We introduce one of the major techniques, namely, the CUmulative SUM (CUSUM) Chart [69]. CUSUM is a sequential technique that assumes that observations are recorded at regular intervals and estimates the point at which the change took place by detecting changes in a parameter, θ, of the data distribution. Given a time series x_1, x_2, x_n at time t_1, t_2, t_n, the approach keeps a cumulative sum of a score $S_i = \Sigma_1^i s_i$at each time point. In an online detection scheme, once the sum of the score S_i exceeds a threshold h, a change can be identified. In the scenario of offline detection, the change point can be identified when the difference between current S_i and the historical minimum S_j is maximized. The score s_i can be defined by the mean, likelihood ratio or standard deviation of the data. For example, the mean based score can be written as $s_i = x(i) - \mu_0$, where μ_0 is the normal mean of the data. Figure 2.8 shows the output of the CUSUM with this setting on the Sahel rainfall index time series data, using an offline manner. The normal mean was estimated using the mean of the entire dataset. As can be seen, the score reaches the maximum at year 1967, marking 1967 as the change point identified by CUSUM. A number of other techniques based on the CUSUM framework appear in the literature as well [70, 71].

Another major technique for change point detection is time series segmentation [72, 73, 74, 75]. For simplicity, we introduce linear segmentation methods. Linear Segmentation algorithms take a time series as an input and return a piecewise linear representation of the time series by approximating it with a number of straight lines. The problem

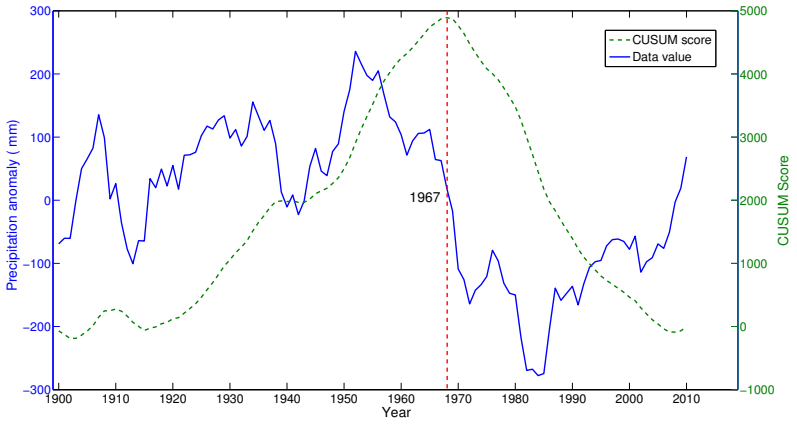


Figure 2.8: A time point change footprint pattern discovered by CUSUM.

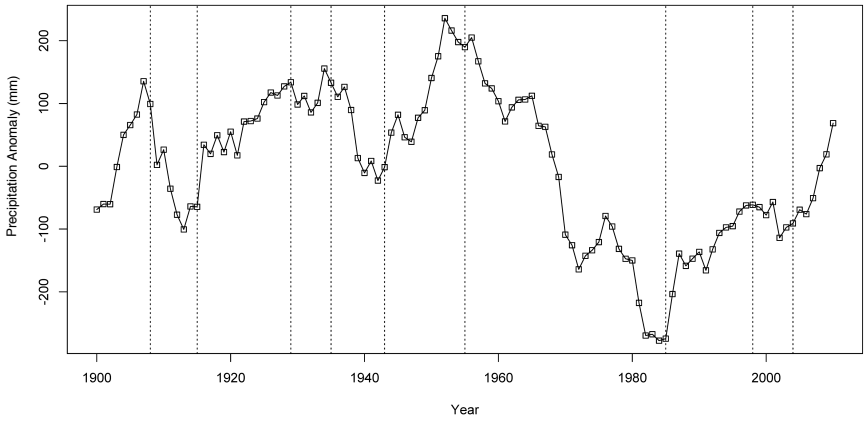


Figure 2.9: Example of time point change identified by time series segmentation.

may be approached in several ways. The sliding window approach starts from the left-most point, attempting to approximate points to the right with a longer segment. If the fitting error exceeds a user specified threshold when adding a new data point to the current segment, the current segment is finalized, and a new segment starts from the new point. This is done repeatedly until the all the data points are examined. Figure 2.9 shows the output of a sliding window approach, which finds 9 change points in a time series. The threshold for adding a new point to the current approximate segment is set to "at most 5 degrees change in slope of the segment". Top-down and bottom-up are two other approaches for time series segmentation. The top-down approach starts with the entire time series and iteratively finds the best change point for the current segment [76, 77]. Bottom-up approaches start with the unit segments and iteratively merge two segments with minimum cost [78].

Time interval change (R1T4): A time interval change pattern is a change in a time series that may last for a number of consecutive time points. In contrast to the abrupt change characterized by the point change pattern, the change interval pattern outlines the duration of rapid or gradual change processes. The time interval change pattern is mostly addressed in a specific domain context, such as climate change and land cover change. For example, the Sahel region in Africa experienced a steady yet fast decrease in rainfall in the late 1960s [79, 80]. Given a time series of annual precipitation in the Sahel region, this decrease over time can be characterized as a change interval starting from 1968 to 1973 where the precipitation dropped 12% in five years.21 This pattern has also been called "abrupt change [28] or an "interval/temporal sub-path of abrupt change [7].

The interesting sub-path discovery technique introduced previously [7] can also be applied on temporal data to find temporal change interval footprints. For example, given a smoothed Sahel rainfall index time series, a "sameness degree interest measure can be applied to a few major intervals with precipitation increase or decrease. Figure 2.10 shows the result of this approach on the smoothed Sahel rainfall index dataset, where a number of time intervals with persistently rapid change are identified.

Patterns with Spatial and temporal footprint

Spatial changes between snapshots (R1T2, R2T2, R3T2): Application domains such as remote sensing and medical image processing are particularly interested in discovering

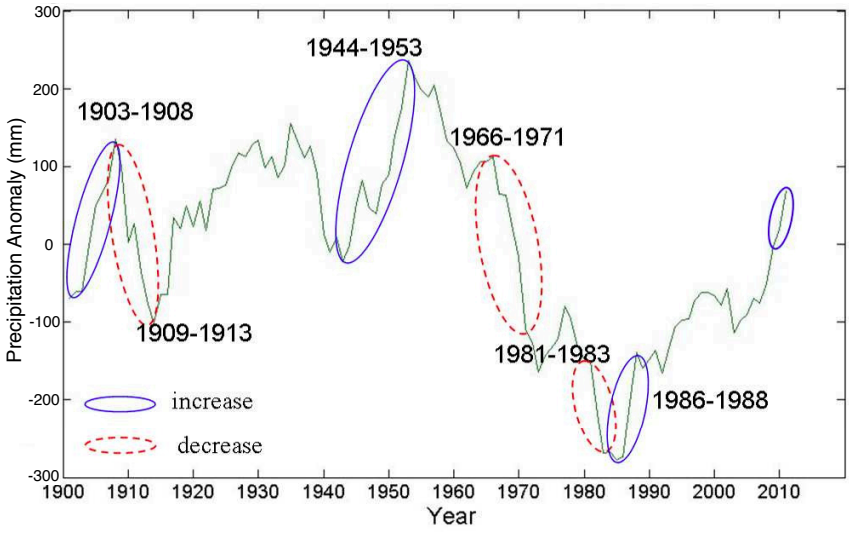


Figure 2.10: Example of interval change on a time series found by the interesting sub-path discovery method.

changes that occur between two or more snapshots (e.g., satellite images) of the same spatial framework. For example, in remote sensing, changes between satellite images help identify land cover change due to human activity, natural disasters, or climate change [81, 82, 83]. This problem is widely studied as the "change detection problem [84, 85, 86, 87, 88, 89, 90].

Given two geographically aligned raster images, the change detection problem aims to find a collection of pixels that have significant changes between the two images. Formally it can be written as:

$$B(x) = \begin{cases} 1, & \text{if there is significant change at pixel } x \\ 0, & \text{otherwise} \end{cases}$$

where B is the desired binary image of decisions [91]. In this definition, a change at a pixel is assumed to be independent of changes at other pixels. We thus classify this pattern as a local change between snapshots (R1T2). In addition, alternative definitions have assumed that a change at a pixel is also associated with its neighborhoods [92]. For example, the pixel values in each block may be assumed to follow a Gaussian

distribution [93]. We refer to this type of change footprint pattern as a focal spatial change between snapshots (R2T3). Researchers in remote sensing and image processing have also tried to apply image change detection to objects instead of pixels [94, 95, 96]. The assumption is that images are composed of homogeneous segments (i.e., objects). A change pattern between two images is defined as a significant difference of an object in the two snapshots. Since the change footprint is a group of pixels (object), we classify this patterns as a zonal spatial change between snapshots (R3T2).

A well-known technique for detecting a local change footprint is simple differencing. The technique starts by calculating the differences between the corresponding pixels' intensities in the two images. A change at a pixel is flagged if the difference at the pixel exceeds a certain threshold. Many methods for selecting this threshold have been discussed in the literature. A variation of this technique known as Change Vector Analysis (CVA) has been used for multi-spectral images [97]. In this case, each pixel is represented by a feature vector with features representing different spectral channels. The difference between the feature vectors of corresponding pixels is used to detect a change at the pixel. Figure 2.13 shows an example output of simple differencing. Dots in Figure 2.11(c)show the locations with changes exceeding 60% of the maximum magnitude between the two images of land surface impervious data [98] shown in Figure 2.11(a) and 2.11(b).

Alternative approaches have also been proposed to discover focal change footprints between images. For example, the block-based density ratio test detects change based on a group of pixels, known as a block [99, 100], rather than a pixel-by-pixel approach. This technique employs hypothesis testing where the null hypothesis corresponds to no change at a given pixel and the alternative hypothesis corresponds to a change. The likelihood ratio is calculated at the pixel and then compared to a defined threshold. If the likelihood ratio exceeds the threshold, the alternative hypothesis is selected; otherwise, the null hypothesis is selected.

Object-based approaches in remote sensing [84, 96, 101] employ image segmentation techniques to partition the image into homogeneous "objects [53]. Classification methods (e.g., decision tree) are then used to classify object pairs in the two images into no-change classes (e.g., water-water) or change classes (e.g., bare land to built-up).

(a) (b) (c)

Figure 2.11: An example of pixel-wise change detection outputs. (a) Impervious surface image of an area in 1986. (b) Impervious surface image of the same area in 1991. (c) Locations with difference exceeding 60% of the maximum magnitude between (a) and (b).

2.4.2 Taxonomy of Change Patterns with Vector-based Spatial Footprint

Spatial footprints on vector data are classified as points (V1), line segments (V2), polygons (V3), and spatial networks (V4). Table 2.2 shows our classification of vector-related change footprint patterns. As can be observed, among the 20 possible combinations, we identified three change footprint patterns that have been explored in the literature: line changes (V2T1), polygon changes (V3T1), and ST volumes (polygon with time interval) changes (V3T4).

Patterns with Purely Spatial Footprint

Spatial lines change (V2T1): Given a set of connected polygons, boundary line segments can be identified such that the attributes in the two adjacent polygons are significantly different. As noted previously, the problem of finding spatial boundaries is commonly known as the spatial Wombling problem. In this particular setting, the problem is known as areal Wombling or polygon Wombling [28]. The spatial line change footprint pattern is of interest to applications such as public health where disease information may be collected in an aggregated format (e.g., total cases in each county) due to confidentiality

Table 2.2: Classification of ST Vector Change Footprints

		Temporal footprint			
		Time point (T1)	Between few snapshots (T2)	Points in Time Series (T3)	Intervals in Time Series (T4)
Spatial footprint	Point(s) (V1)				
	Line segments (V2)	Find boundary lines separating areas with high and low risk of diseases (V2T1)			
	Polygon (V3)	Find the regions where the risks of disease is higher inside than others (V3T1)			Find a region and a time interval where the risk of disease is increasing during this interval (V3T4)
	Spatial network (V4)				

requirement [28]. Areal Wombling helps find boundaries separating high-risk and low-risk counties.

Areal Wombling can be done in a similar way as lattice Wombling. The change magnitude at each boundary line can be computed as the difference of attribute values between adjacent polygons. The changes exceeding a threshold or in the top k% form the output footprint. Figure 2.12 shows an example of boundaries highlighting significant difference of national GDP growth in 2011 [37]. It is done by selecting the boundaries with the top 20% highest difference in GDP growth between the two neighboring countries.

Statistical approaches have been designed to evaluate the significance of these footprints. For example, in a hierarchical Bayesian approach [28], a boundary likelihood value (BLV) is estimated for each boundary using the Markov Chain Monte Carlo (MCMC) method. Similar approaches have been developed on geostatistical [27] and point process [102] data models.

Spatial polygon change (V3T1): A change with a polygon footprint delineates a region where some attributes have significantly changed. For example, given a spatial

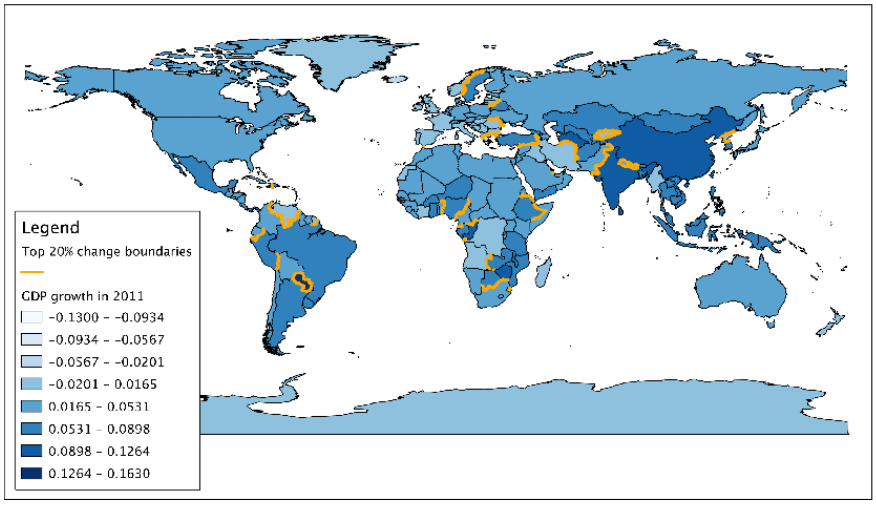
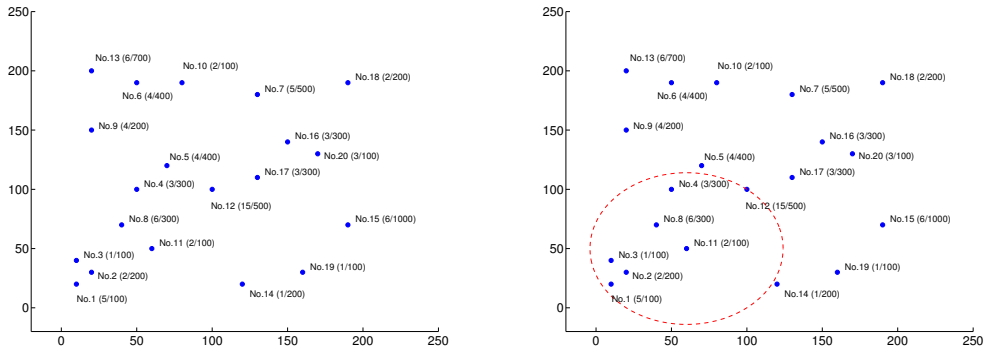


Figure 2.12: Change boundary (line) footprints on the world GDP growth data (best viewed in color).

point process dataset representing disease cases, a spatial polygon change pattern draws a polygon in which the density of points (disease) inside the polygon is significantly higher than outside. Such patterns can also be identified on a set of polygons (representing geopolitical regions) where the total disease count of each polygon is specified. This problem is referred to as spatial cluster detection [103]. In public health/epidemiology, finding spatial clusters with a higher density of disease is of great interest to understand the distribution and spread of disease.

Kulldorff et al. proposed a spatial scan statistics framework [103] for disease outbreak detection. Given a number of fixed locations (e.g., hospitals) and the number of disease cases at each location, the early version of this work focuses on finding the most likely spatial region where the relative risk of disease inside is higher than outside. The spatial scan statistics employs a likelihood ratio test where the null hypothesis H_0 is that the probability of disease inside a region is the same as outside the region, and the alternative hypothesis H_1 is that there is a higher probability of disease inside than outside. Assuming that the total disease count in a region follows a Poisson model, the likelihood ratio can be formally written as: $\frac{(\frac{n_z}{\mu_z})^{(n_z)}(\frac{n_G-n_z}{\mu_G-\mu_z})^{(n_G-n_z)}}{(n_G/\mu_G)^{(n_G)}} I(\frac{n_z}{\mu_z} > \frac{(n_G-n_z)}{(\mu_G-\mu_z)})$, where n_G and n_z are the number of observations in the entire area and in candidate region Z respectively, and μ_G and μ_z are the expected number of disease reports in the entire area and the region Z. The expected numbers may be derived from a baseline population at risk or estimated using non-parametric models according to the size of the region. All the spatial regions represented by a circle or ellipsoids in the spatial framework are enumerated and the one that maximizes the likelihood ratio score is identified as the candidate. Finally, the statistical significance of the candidate cluster is tested by a Monte Carlo simulation. Figure 13(a) shows an example of spatial point dataset where each point represents a fixed location (numbered 1-20). The number of disease cases and population in each location are labeled (i.e., cases/population). The most likely cluster outlined by the red circle is shown in Figure 13(b), assuming the data follows a discrete Poisson model. As can be observed, locations in the circle have a higher rate of disease compared to the outside (about 1/100).

Later extensions on the same ideas have explored normal [104], exponential [105],



(a) (b)

Figure 2.13: An example of pixel-wise change detection outputs. (a) Impervious surface image of an area in 1986. (b) Impervious surface image of the same area in 1991. (c) Locations with difference exceeding 60% of the maximum magnitude between (a) and (b).

ordinal [106], and non-parametric models [107]. The scan statistic has also been generalized to handle spatiotemporal point (event report) datasets [108, 109] and irregular-shaped clusters [110]. Computational efficiency of the scan statistic method was further optimized by employing a top-down pruning of the search space [111]. A Bayesian version of scan statistics uses an inference instead of the frequentist hypothesis testing [112]. SaTScan [113] is a software tool developed for discovering spatial and spatiotemporal clusters which integrates the above models.

Patterns with Spatial and Temporal Footprint

ST volume change footprint (V3T4): The ST volume change footprint represents a change process occurring in a spatial region (characterize by a polygon) during a time interval. It quantifies both the spatial coverage and the temporal duration of a non-stationary ST process. For example, an outbreak event of a disease can be defined as an increase in disease reports in a certain region during a certain time window up to the current time. Change patterns known to have an ST volume footprint include the ST scan statistics (introduced above) and emerging ST clusters defined by Neill et al [114].

Given an ST point process dataset, and a baseline risk probability p, an emerging

cluster is defined as a spatial region S and a time window starting at t_{min} such that the risk of each instance in S in days $t_{min}, t_{min+1}, , T$ is monotonically increasing: $q_i \times p$, where $1 \leq q_{min} \leq q_{min+1} \leq \ \leq q_T$. The null and alternative hypotheses are defined as follows: H_0: the probability that one instance is at risk in the cluster for each day are the same and equals the expected value p, and H_1: The above probability in days $t_{min}, t_{min+1}, , T$ is $q \times p$ where $1 \leq q_1 \leq q_2 \leq \ \leq q_T$. The following likelihood ratio is tested over all the spatial regions S and start time $t_m in$ pairs: $\frac{MAX_{1 \leq q_{tm\,in} \leq q_T} \Pi q_t^{C_t} \cdot e^{-q_t B_t}}{e^{-B}}$, where C_t and B_t are the total observations and total expected number (baseline) of disease reports respectively, in day t in S, and B is the total number of expected diseases (baseline) for the entire window in S. The most likely pair (S, t_{min}) is the output pattern.

2.5 Future Directions and Research Needs

An important goal of our change footprint taxonomy was to uncover gaps in the research. In this section, we discuss the change footprint patterns that have yet to receive much, if any, attention by researchers. Some may be interesting to data scientists and could be explored in the future. To highlight interesting vector change footprint patterns, we show two case studies from historical GIS applications at the end of this section. We believe that they represent exciting examples of avenues for future research.

2.5.1 Raster-based Patterns

Table 2.1 shows that most of the raster footprint patterns have been explored. However, we did identify four footprint patterns that may be further investigated, i.e., spatial focal and zonal changes that occur at a time point or a time interval in time as shown in Table 2.3. Hence we suggest some change phenomena that appear to exhibit such footprints.

Spatial focal change at a time point: A spatial focal change at a time point can be modeled in a spatial time series dataset. One example of a spatial focal change at a time point may be described as follows: given a spatial time series dataset, find a location s and a time point t where the time series at s starts to behave very differently from time series in its spatial neighborhood at t. This pattern may be of interest to

Table 2.3: A full list of raster ST change footprint patterns

		Temporal footprint			
		Time point (T1)	Between few snapshots (T2)	Points in Time Series (T3)	Intervals in Time Series (T4)
Spatial footprint	Local	R1T1 (N/A)	Remote sensing image change detection (R1T2, R2T3, R3T2)	Change point detection (e.g., CUSUM) R1T3	Change interval discovery on time series R1T4
	Focal	Edge Detection (e.g., Lattice Wombling) R2T1		Find a location where the time series changes differently than its spatial neighbors at a time point. R2T3	Find a location where the time series changes differently than its spatial neighbors during a time interval. R2T4
	Zonal	Interesting sub-path discovery (e.g., ecotones) R3T1		Find a region where the aggregate/summary time series has a change at a time point. R3T3	Find a region where the aggregate time series has a persistent change during a time interval R3T4

research in climate science and remote sensing. For example, given a spatial time series dataset representing land cover indices, one may be interested in finding locations and the corresponding time point of an abrupt land cover change due to forest fire or other human introduced events. Due to spatial autocorrelation, we know that neighboring time series tend to behave similarly in normal scenarios. A focal change in space can distinguish a time series from its neighbors in the case of anomalous events, even if the change on the time series itself is not very abrupt. Similarly, we can define a spatial focal change in a time interval as an event that a time series exhibits behavior significantly different from its spatial neighbors during a contiguous time period. Spatial zonal change at a time point/in an interval: This pattern may be characterized as an abrupt change (point) or rapid/gradual change (interval) that occurs on the aggregated time series of a spatial region. This pattern may be explored to address problems such as finding regional climate change patterns, and automatic finding of large scale of land cover change, etc.

2.5.2 Vector-based Patterns

Compared to raster ST change footprint patterns, vector ST change patterns have been much less intensively explored. The existing patterns ahaond techniques focus mostly on finding changes in non-ST attributes (e.g., count of disease) with vector spatial footprints. We believe that there is also great value in understanding the change patterns of ST attributes of vector spatial objects. For example, attention may be given to spatial change patterns with network footprints (V4T1). In Table 2.4 we list numerous examples of questions users may pose in a variety of applications in addition to those listed in Table 2.2.

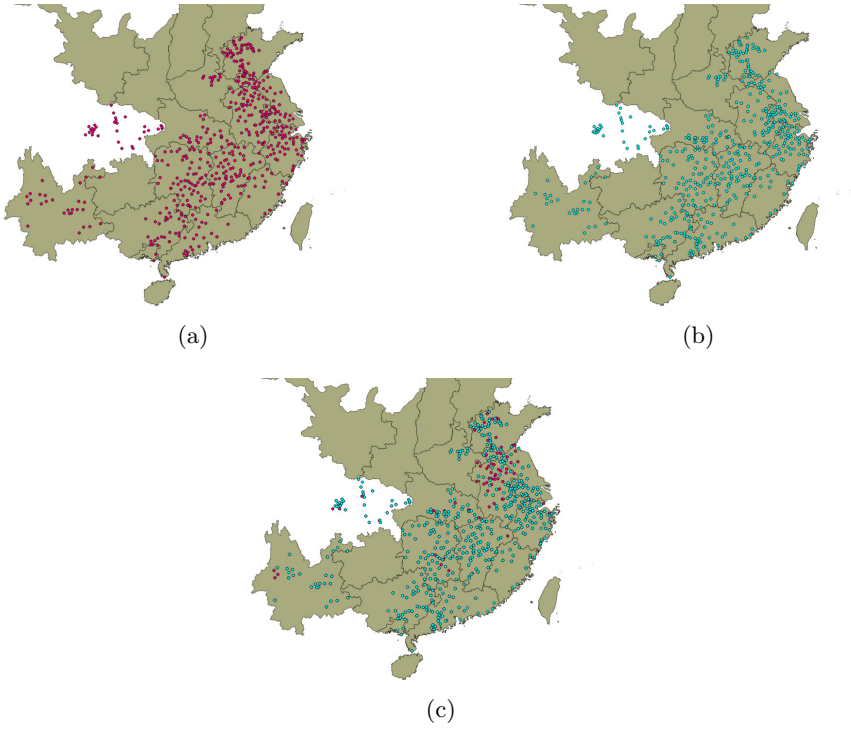
2.5.3 Change Footprint Patterns in Non-Scientific Domains

Change footprint patterns are valuable to study even in domains far removed from scientific domains. One field where it is critical to understand how spatial relationships change over time is the study of history [115]. Such changes can be modeled as ST change with vector spatial footprints. Specifically, we illustrate two application examples: the China historical jurisdiction change footprint patterns and the U.S. railroad change footprint patterns.

Case study 1: Historians studying Chinese history to trace changes in administrative hierarchies down to the "Xian(county) level from 221 BC to 1911 AD. The county seats (capital) in Chinese history are usually used to represent the real location of the counties. Vector datasets representing the county seats at different times are available from the China Historical GIS project [116, 117]. In this dataset, each historical county seat is represented as a spatiotemporal point, where the location, valid time period, and non-ST attributes (e.g., name) are available. For example, Figure 2.14 show different snapshots of the county seat locations in 300 AD and 400 AD, and the difference between the two years in Chinese history.

Given such a ST vector dataset, one may define a few change patterns with vector ST footprints. For individual points, one may identify the time period with frequent change of location. A regional pattern (e.g., polygon) summarizing a large number of location shifts during a given time period (e.g., 300 AD to 400 AD) can be identified to indicate the dynamic of the geo-political relationships in this area. For example, a

significant change in point locations can be found in the east of the map as shown in Figure 2.14(c).



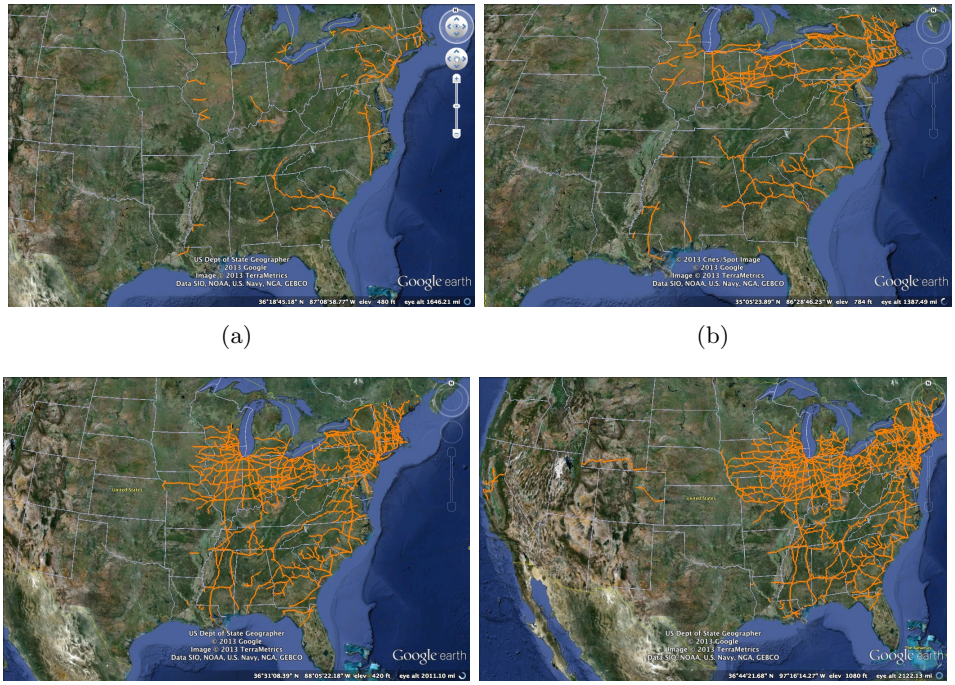
(a) (b)

(c)

Figure 2.14: A comparison of county seats locations at 300 and 400 AD in China (best viewed in color). (a) County seats of China in 300 AD. (b) County seats of China in 400 AD. (c) Change of county seat locations between 300 AD and 400 AD.

Case Study 2: Railroad growth in the 19th century is another subject of interest to historians. The expansion of the U.S. railroad network contributed significantly to the immigration and employment in the country. The complexity and scale of railroad operations and their uneven extension across the landscape created both intensive and extensive changes in American communities [118]. Understanding the change pattern of the railroad network helps understand U.S. social and economic development in the 19th century [119]. A historical railroad network dataset is available for this study [120]. Figure 15 shows a few snapshots of this datasets in 1840, 1850, 1861, and 1870.

Given such a ST vector dataset, one can define change patterns with line/network



(a) (b)

(c) (d)

Figure 2.15: U.S. railroad network expansion in the 19th century (best viewed in color). (a) U.S. railroad network in 1840. (b) U.S. railroad network in 1850. (c) U.S. railroad network in 1861. (d) U.S. railroad network in 1870.

spatial footprints. One possible pattern may be a time interval during which the spatial network grew significantly (e.g., total length, coverage) in a particular area (e.g., Midwest). For example, in the snapshots a significant expansion of the network can be identified between 1840 and 1851. Another pattern that could be described is the directions in which the spatial network expanded during a certain time period (e.g., 1840-1851). For example, in the given dataset, the network expands towards the west (e.g., Midwest) and southwest between 1840 and 1870.

2.6 Summary

This chapter proposes a taxonomy of spatiotemporal change footprints that may be of use to researchers across multiple research domains. We built the taxonomy after conducting a multi-disciplinary review of research in ST change pattern discovery. Our taxonomy achieves two valuable goals. First, it classifies a wide variety of ST change footprints that have already received attention in different domains. Previously, much of this research was hidden from view, so to speak, due to the lack of common terminology across disciplines for discussing similar phenomena. Second, our taxonomy reveals gaps in the research, that is, change footprint patterns that have yet to be studied despite their potential applicability to many real-world problems. We especially note the need for research on ST change footprints on vector data.

In the future, we plan to incorporate other aspects of change patterns in our classification. Currently, the taxonomy mainly focuses on only univariate (single non-ST attribute) techniques. The next step will be to include multivariate definitions and approaches. Also, we want to address issues such as computational structure and statistical modeling of change patterns.

Table 2.4: A full list of vector ST change footprint patterns.

		Temporal footprint			
		Time point (T1)	Between few snapshots (T2)	Points in Time Series (T3)	Intervals in Time Series (T4)
Spatial footprint	Point(s) (V1)	N/A	Find significant shift of locations of a point process at two different time (V1T2)	Which county seats abruptly change their location? When? (V1T3)	Find county seats that change their locations frequently during some time period in Chinese history. (V1T4)
	Line segments (V2)	Find boundary lines separating areas with high and low risk of diseases (V2T1)	What is the difference in direction of a road/river after earthquake? (V2T2)	When and where did a person change his/her route to work? (V2T3)	Whne & where did the Mississippi river significantly change its route in the last century? (V2T4)
	Polygon (V3)	Find the regions where the risks of disease is higher inside than others (V3T1)	Which county in China has changed its area significantly between 800 AD and 900 AD (V3T2)?	Which bird habitats significantly shrank their area due to urban sprawl? When? (V3T3)	Find a region and a time interval where the risk of disease is increasing during this interval (V3T4)
	Spatial network (V4)	Find the subnetwork of the road networks where the risk of crime becomes higher than other parts (V4T1)	Find the subnetwork that summarizes the change between the two snapshots of the network (V4T2)	Find a time/places when/where the railroad network significantly grows (V4T3)	Find the expanding direction of the network during a time period (V4T4)

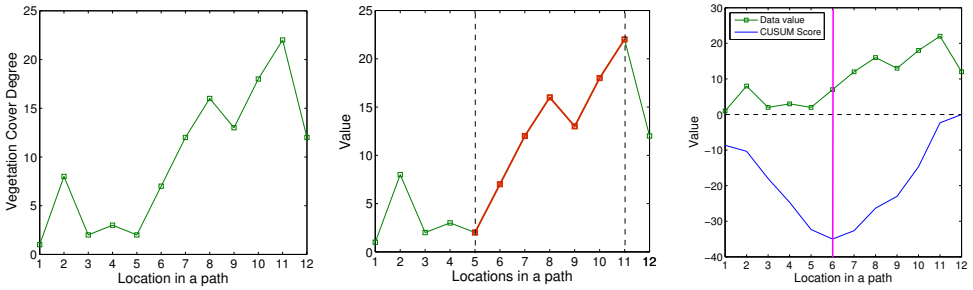
Chapter 3

Discovering Interesting Sub-path in Spatiotemporal Datasets

3.1 Introduction

A spatiotemporal (ST) field data model [36]consists of two parts, a spatiotemporal framework, which is a partition of space or time (e.g., latitude and longitude grid), and a function defined on the framework (e.g., the degree of vegetation cover). Given a spatiotemporal (ST) dataset and embedded in its spatiotemporal framework, the goal of the interesting sub-path discovery problem is to identify all qualifying contiguous subsets of the given path according to an interest measure. For example, Figure 3.1(a) shows the degree of vegetation cover along a particular spatial path (a highway or a longitude across the continent). Given an interest measure of abrupt change, one may identify a sub-path from location5 to 11. Vegetation cover in this sub-path exhibits an abruptly increasing trend as shown in Figure 3.1(b). Given an interest measure of stability (e.g., low standard deviation), we alternatively find the sub-path from location 3 to location 5, where a relatively stable trend can be observed.

The ability to discover interesting sub-paths is important for many application domains. For example, vegetation cover is often used to study the response of ecological zones to climate change, which may vary across different ecological zones in the world. Given a path (e.g., along a longitude in Africa) and an interest measure of abrupt change, one can find sub-paths (e.g., the paths across the Sahel) with sharp increases



(a) Vegetation cover along a path (b) An interesting sub-path in the path (c) An interesting point found by CUSUM

Figure 3.1: An example of interesting sub-path in the data and found by related work (best viewed in color).

(decreases) of vegetation cover. Such sub-paths may outline the spatial footprint of the transitional areas (known as ecotones [121]) between ecological zones. Due to their vulnerability to climate changes, finding and tracking ecotones gives us important information about how the ecosystem responds to climate changes. Interesting sub-path discovery also contributes to other applications. Water quality monitors may be interested in river sub-paths where water quality changes abruptly. State traffic engineers may be interested in highway sub-paths where traffic speed is unstable (e.g., with high standard deviation). Coastal area authorities may be interested in coast line sub-paths which are prone to rapid environmental change due to rising ocean and melting polar icecaps.

Discovering interesting sub-paths is challenging due to the following reasons: First, the lengths of the sub-paths of interest may vary, and have no predefined maximum length. For example, the length of flood-prone sub-paths along long rivers (e.g., the Gange, Mississippi, etc) may extend hundreds or thousands of miles. Second, the interestingness in a sub-path may not exhibit monotonicity, i.e., uninteresting segments may be included in an interesting sub-path. For example, the interesting sub-path from location 5 to 11 in Figure 3.1(a) exhibits non-monotonicity from location 8 through 9 with respect to the interest measure of abrupt increase. Third, the data volume is potentially massive. For example, consider the problem of finding all the interesting longitude sub-paths exhibiting abrupt change in an eco-climate dataset with attributes

such as vegetation, temperature, precipitation, etc., over hundreds of years from different global climate models and sensor networks. The volume of such datasets will range from terabytes to petabytes.

Previous work on interesting spatiotemporal sub-path discovery has focused on change point detection using one dimensional or two dimensional approaches, as shown in Figure 3.2. One dimensional approaches aim to find points in a temporal path where there is a shift in the data distribution [69, 122, 123, 18, 124]. For example, Figure 3.1(c) shows the output of the CUSUM [69, 70] approach on the sample data in the previous example. It finds location 6 as a point of interest (with abrupt change from below the mean to above the mean). Two dimensional approaches, such as edge detection [125], find boundaries between different areas in an image. Given a linear path, this technique will identify a change point at the intersection of an edge and the given path.

The above related works are limited to detecting points of interest in a ST path, rather than finding long interesting sub-paths. In contrast, this work proposes a novel computational framework to discover sub-paths of arbitrary length based on a given interest measure.

In our early work, we proposed a sub-path enumeration and pruning (SEP) approach, with two design decisions, namely, the Bottom-up traversal after leaf-evaluation with in-row pruning (BURP, perviously named the row-wise strategy), and the Breadth-first-traversal-after leaf-evaluation with sub-graph pruning (BSGP, previously named the top-down strategy). The second approach outperforms the first for longer patterns but brings $O(n^2)$ memory cost.

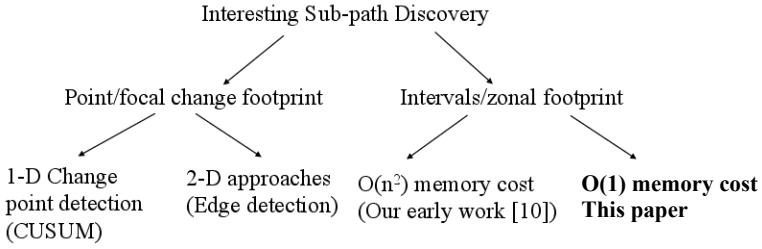


Figure 3.2: A classification of related work.

In our early work [7], we made the following contributions: (1) We proposed and formulated the general interesting ST sub-path discovery problem. (2) We proposed a

novel sub-path enumeration and pruning (SEP) approach with two designs with different traversal orders, namely, a bottom-up traversal after leaf-evaluation with in-row pruning (BURP, previously named as SEP row-wise strategy), and a Breadth-first traversal after leaf-evaluation with sub-graph pruning (BSGP, previously named as SEP top-down strategy), to discover the defined sub-paths. (3) We provided theoretical analysis of the correctness, completeness and computational complexity of the proposed approach. (4) We defined one specific type of interesting sub-path, namely sub-paths of abrupt increase (decrease), with a novel interest measure "sameness degree". We showed through case study results that the SEP approach discovered interesting spatial and temporal sub-paths representing ecotones and abrupt climate changes, (5) We provided an experimental evaluation of the proposed SEP approach using large test datasets to compare the performance and scalability of the two proposed design decisions of SEP. Results showed that for longer patterns, the top-down strategy outperforms the row-wise strategy, and both are orders of magnitude faster than the naive approach.

This work proposes a new algorithm, the Row-wise after leaf-evaluation with Column Pruning (RTCP), which traverses the space of all sub-paths (i.e., the G-DAG in [7]) efficiently with $O(1)$ memory cost. Table 3.1 summarizes the advantages of the new RTCP algorithm over the two previous algorithms. The new RTCP G-DAG traversal algorithm significantly reduces the memory cost while still visiting minimum number of nodes in the graph. The time cost of the RTCP is also much smaller than the previous algorithms due to fewer memory accesses for book-keeping.

Specifically, we make the following additional contributions in this work: (1) We formally model the interesting sub-path discovery problem as a graph traversal problem, and formulate our previous algorithms as G-DAG traversal strategies, namely, Bottom-Up traversal after leaf-evaluation with in-Row Pruning (BURP), and the Breadth-first traversal after leaf-evaluation with Sub-Graph Pruning (BSGP). (2) We propose a new G-DAG traversal algorithm, namely, the Row-wise Traversal after leaf-evaluation with Column-Pruning (RTCP) algorithm, which traverses the G-DAG in a row-wise manner with a pruning border on each column of the G-DAG. (3) We provide theoretical analysis of the correctness, completeness and computational complexity of the proposed RTCP algorithm. (4) We validated RTCP's computational efficiency over the two previous approaches on large synthetic and real datasets. Results show that RTCP dominates

the previous design decisions in both the best and the worst scenarios.

Scope: This chapter deals with interesting sub-paths given a linear path such as a costal river line, a highway, a trajectory, etc. It does not deal with sub-regions or sub-volumes. Characterization of the statistical distributions of properties of sub-paths of different lengths is also beyond the scope of this chapter. The technique proposed is evaluated using case studies of abrupt climate change. We do not, however, address detailed issues in abrupt climate changes. Also beyond the scope of this chapter is detailed discussion of trajectory mining and mobility patterns.

Outline: The rest of the chapter is organized as follows: Section 3.2 introduces basic concepts and formalizes the interesting ST sub-path discovery problem. Section 3.3 introduces our contributions. We first propose a G-DAG traversal framework with two steps, namely, leaf-evaluation and G-DAG traversal, as the computational solution to our problem, and then formulate the previous SEP design decisions as G-DAG traversal algorithms. Then we present our new Row-wise after leaf-evaluation with Column-Pruning (RTCP) algorithm. Theoretical analysis of the G-DAG traversal algorithms is presented in Section 3.4. Section 3.5 shows results of experimental evaluation. Section 3.6 defines an instance of an interesting sub-path and presents two case studies on real eco-climate datasets. In Section 3.7 we discuss the difference between this work and some other problems involving ST paths, such as trajectories and time series. Section 3.8 finally concludes the chapter with possible future work.

Table 3.1: A comparison of G-DAG traversal strategies (after leaf-evaluation) in this work and previous work

	Preliminary work [7]		**Proposed work**
	BURP (previous SEP row-wise)	BSGP (previous SEP top-down)	**RTCP**
A: Avoid unnecessary visits to dominated non-leaf nodes	No	Yes	**Yes**
Memory need (n = numbers of units in the path)	$O(n)$	$O(n^2)$	**O(1)**

3.2 Problem Formulation

This section introduces some basic concepts used in the formulation of the interesting sub-path discovery problem. We then give the formal problem statement, with an illustration using an example application problem.

3.2.1 Basic Concepts

A simple spatiotemporal path is a collection of contiguous locations or time points in a spatiotemporal field. Examples of simple spatiotemporal paths include a longitude, a trajectory, or a time series, etc.

Definition 1. ***Sub-path:*** *Given a simple spatiotemporal path S consisting of N locations $s_1, s_2, ..., s_N$, a sub-path $W = (i, j)$ of S is a contiguous subset of locations from s_i to s_j in S.*

A **unit sub-path** is the smallest sub-path which contains two contiguous locations. A value is associated with each unit sub-path representing some attribute of this sub-path. The attribute can be defined based on the specific problem. For example, if we aim to find sub-paths related to changes, the value associated with each unit sub-path is likely to be defined as the difference between the two neighboring locations. The number of unit sub-paths $n = N - 1$. Figure 3.3 shows the input path in the given example, with the unit sub-paths being the ones between consecutive locations (e.g., (1, 2), (5, 6), etc). The corresponding difference between locations are also shown, as the associated value for each unit sub-paths. Next we introduce the concept of interest measure of a sub-path.

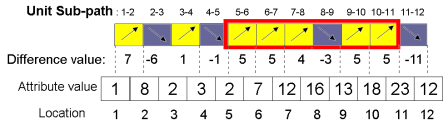


Figure 3.3: An example ST path

Definition 2. ***Sub-path Interest measure:*** *An interest measure of sub-path W is an aggregate function which takes values associated with each unit sub-path in W and returns a numerical result representing the extent of interestingness of the sub-path.*

Formally, the function can be written as $F_{spi} : R^n \rightarrow R$*, where* n *is the total number of unit sub-paths in* S*.*

An interest measure can be the mean, max, standard deviation, or any other complex statistic of a list of values. For the example shown in Figure 3.1, the interest measure can be defined as "the average of value differences in each unit sub-path". This is equivalent to the "slope" of each sub-path. The interestingness of the sub-path 5-11 is thus 3.5.

Aggregate functions can be categorized as distributive, algebraic or holistic [126]. Distributive functions are those that can be computed by a linear scan with only one temporary variable, such as *sum*, *count*, etc. Algebraic functions can be computed by a constant number of distributive functions, such as *average*, which can be computed by *sum/count*. Holistic functions are those which cannot be computed using a constant number of distributive functions, such as *median*. In this chapter we mainly focus on interest measures that are algebraic functions (e.g., standard deviation) as is the case for most statistical tests. This type of interest measure can be computed using a constant number m of distributive functions $\{D^1_{aggr}, D^2_{aggr}, ..., D^m_{aggr}\}$.

Definition 3. ***Interesting Sub-Path (ISP):*** *Given a ST path* S*, an aggregate function* F_{spi} *as interest measure, and an interestingness test function* $T : R \rightarrow \{True, False\}$*, a sub-path* $W = (s_i, s_j)$ *is an interesting sub-path (ISP) of* S *if* $T(F_{spi}(W)) \rightarrow True$*.*

For example, in the sample data path given in Figure 3.3, we consider a sub-path to be interesting if "the average value difference in each unit sub-path is at least 3.5". By this simple measure, we easily find that sub-path (5, 11) is an ISP.

According to our definition, part of an ISP may also be an interesting sub-path (e.g., sub-path 5-7 in the above example). Identifying both a super-path and a sub-path is redundant. In addition, in real applications, longer sub-paths are usually more important than shorter ones to the understanding of the underlying pattern. Thus, we define a dominant interesting sub-path (DISP) as follows:

Definition 4. ***Dominant Interesting Sub-Path (DISP):*** *Given a ST path* S *and an interest measure function, a dominant interesting sub-path (DISP) is an interesting sub-path (ISP) of* S *that is not a subset of any other ISP. We say sub-path* A *"dominates" sub-path* B *if* B *is a subset of* A*.*

For example, in the previous example, sub-path 5-7 is an ISP (interest measure 5.0). However, it is a subset of ISP 5-11 (interest measure 3.5) and thus not a DISP. Other constraints on interesting sub-paths may be added to meet specific domain requirements or for simplicity (e.g., ISPs should start and end with certain locations). Specific design decisions can be applied in the discovery process to reduce the size of the search space while preserving the dominating relationship.

3.2.2 Problem Statement

The interesting sub-path discovery problem can be formally expressed as follows:

Given:

- A path S in a ST framework containing n unit sub-paths
- A function f of values associated with unit sub-paths
- An aggregate function measuring sub-path interestingness $F_{spi} : R^n \rightarrow R$
- An interestingness test function $T : R \rightarrow \{True, False\}$

Find:

- All dominant interesting sub-paths in S

Objective:

- Reduce computational cost

Constraint:

- Correctness of the results, i.e., all the sub-paths found by the solution should be DISP
- Completeness of the results: all the sub-paths that are DISPs must be found by the solution
- F_{spi} is an algebraic aggregate function

The test function T is applied on the result of the interest measure function F_{spi}, as discussed in Definition 3. For the given input data shown in Figure 3.1, the interest measure is "the average value difference of each unit sub-path," and the test is "at least 3.5." The corresponding output DISP will be sub-paths 1-2 and 5-11, with interest measures 7.0 and 3.5 respectively. There also could be other constraints on the DISPs, such as the DISP should start and end with interesting unit sub-paths, etc. We show an example with such constraints in Section 3.6 to filter the meaningful results.

3.3 Computational Approaches

A sub-path enumeration and pruning (SEP) approach was proposed in our previous work [7]. In this section, we review the key ideas of the SEP approach, and formulate the computational structure of the Interesting Sub-path Discovery problem as a graph traversal framework with two components, namely, leaf-evaluation, and a G-DAG traversal strategy after leaf-evaluation. Then we propose a new G-DAG traversal strategy, Row-wise Traversal (after leaf-evaluation) with Column Pruning (RTCP) which has a lower time and space cost.

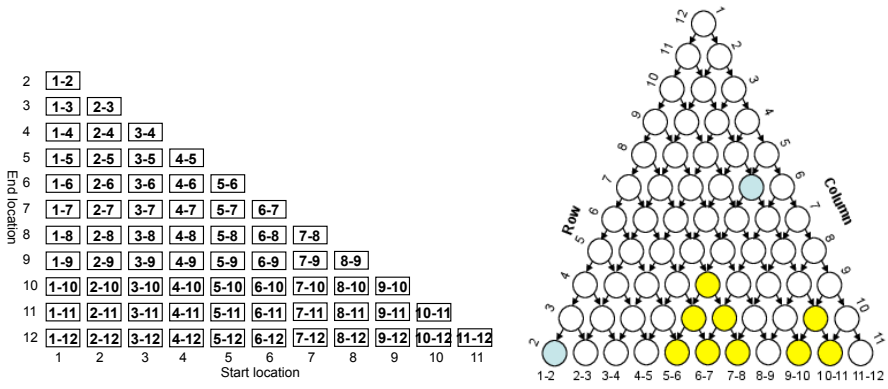
3.3.1 A Graph Traversal Framework

The set of all the sub-paths in a path can be represented as a grid structure based on each sub-path's start and end location. Figure 3.4(a) shows this grid representation for the example path in Figure 3.1 where each row contains sub-paths with a common start location. The cells along the diagonal are the unit sub-paths. As can be seen, each cell (sub-path) is dominated by all the cells to the left-bottom quadrant of it. Adding the "dominating" relationship among the neighboring cells, we get a directed acyclic graph, where each node is a sub-path in S and each directed edge is a dominating relationship between a pair of sub-paths. We call it grid-based directed acyclic graph (G-DAG). Figure 3.4(b) shows a G-DAG for the example path in Figure 3.1, whose sub-paths and dominance relationships are given in Figure 3.4(a).

We define several properties of a G-DAG as follows.

Definition 5. *Each node in a G-DAG has a row and a column number. The column number equals the start location of the sub-path it represents in the underlying dataset, while the row number equals the end location of the sub-path it represents in the data. For example, sub-path 5-11 is on row 11, column 5. For a spatial path with n locations, there are* $n \cdot (n-1)/2$ *sub-paths. Correspondingly, in the G-DAG representation of this spatial path, there are* $n \cdot (n-1)/2$ *nodes.*

Definition 6. *Each inner node in a G-DAG has two parent nodes (direct predecessor), which are sub-paths longer by 1 unit. Nodes along the two borders of the G-DAG have only one parent. The root node has no parent. For example, the two parents of node (5, 11) are nodes (5, 12) and (4, 11).*



(a) A grid representation of all the sub-paths (b) Grid-based DAG

Figure 3.4: An illustration of the enumeration space and corresponding grided-DAG representation (best viewed in color).

The computational solution to the interesting sub-path discovery problem thus can be summarized as "Traverse the G-DAG representation of the input ST path to find dominant interesting sub-paths (DISP)." There are two challenges in this process: (1) In order to evaluate each node, we need to compute the interest measure by scanning the corresponding sub-path. This may introduce repeated scans of the same sub-paths and increase the computational cost. (2) We need an efficient order to traverse the G-DAG. To address these challenges, we propose a two-phase G-DAG traversal framework. Phase 1 a leaf-evaluation of the G-DAG for running tally pre-computation. Phase 2 traversal on the entire G-DAG. We illustrate the two phases in the following subsections.

3.3.2 Approach to Challenge 1: leaf-evaluation

One solution to the first challenge is to materialize a lookup table. To reduce the computational cost, the goal is to achieve a constant-cost computation of an aggregate function over any sub-path using the table, and limit the computational cost of building such a table to O(n). In a G-DAG, all the attribute values are associated with the leaf nodes. Building the lookup table can be implemented as a linear evaluation of all the leaves in the G-DAG.

As defined previously, the interest measure function F_{isp} is an algebraic aggregate

Sub−path	(1,2)	(1,3)	(1,4)	(1,5)	(1,6)	(1,7)	(1,8)	(1,9)	(1,10)	(1,11)	(1,12)
SUM	7	1	2	1	6	11	15	12	17	22	12

Figure 3.5: Lookup table for the SUM function in the sample data.

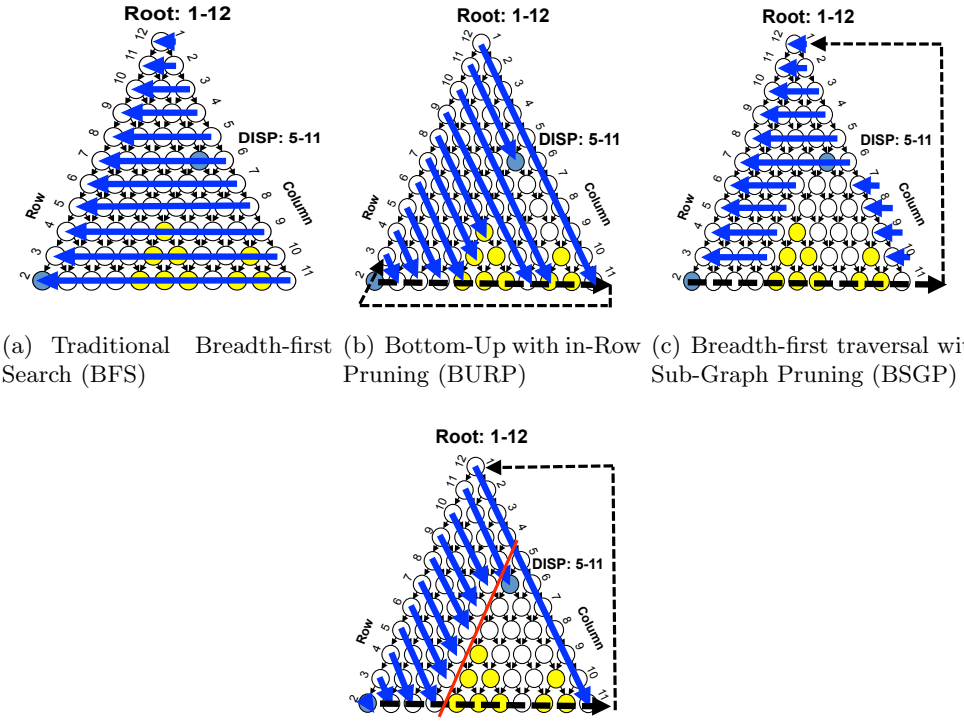
function (e.g, average), which can be decomposed into a constant number of distributive aggregate functions (e.g., sum, count). We build one lookup table for each distributive function by making use of the distributive property. For example, we establish a table for the sums from unit sub-path 1 to each unit sub-path k by a sequential evaluation of all the leaf nodes in the G-DAG (values associated with each each unit sub-path). As a result, the sum of values in any sub-path (i, j) can be computed using $sum(1, j) - sum(1, i)$. Figure 3.5 shows the lookup table of SUM function over the sample ST path given in Figure 3.3. For example, the sum of values in sub-path (3,5) can be computed using SUM(1,5)-SUM(1,3)=0. In other words, we are computing the aggregation of a sub-path W based on its super-path W' and the corresponding complement sub-path W'-W. Unfortunately, not all distributive aggregate functions have such a property. For example, it is obvious that min and max of a sub-path cannot be computed in this way. Functions having this property are known as reversible aggregate functions [127]. In this chapter, we assume that all the distributive aggregate functions used in the interest measure are reversible (which is true for most of the statistical functions).

By doing the leaf-evaluation, we make sure that each node can be evaluated at constant time cost during a G-DAG traversal. In the rest of the section, we focus our discussion on how to design efficient G-DAG traversal strategies, which is the core part of the solution.

3.3.3 Approach to Challenge 2: Efficient G-DAG Traversal

In this section, we present three G-DAG traversal strategies to tackle the second challenge named in Section 3.3.1. We first formally define the concept of a G-DAG traversal strategy, and briefly compare different traversal strategies. Then we present detailed algorithms of these traversal strategies. All the algorithms are designed based on the assumption that a lookup table has been built via leaf-scan on the G-DAG.

Definition 7. ***G-DAG traversal strategy:*** *A G-DAG traversal strategy is an order to examine the nodes in a G-DAG such that all the DISP nodes (those corresponding to*



(a) Traditional Breadth-first Search (BFS)

(b) Bottom-Up with in-Row Pruning (BURP)

(c) Breadth-first traversal with Sub-Graph Pruning (BSGP)

(d) Row-wise Traversal with Column Pruning (RTCP)

Figure 3.6: An illustration of G-DAG leaf-evaluations and traversal strategies(best viewed in color).

dominant interesting sub-paths) can be identified.

A simple example of a traversal strategy can be a traditional Breadth-First Search (BFS). As shown in Figure 3.6(a), the traverse starts from the root nodes of the G-DAG and expands to both children of each node iteratively. If a node is DISP then it is added to the candidate list. As a result, the BFS traversal strategy will examine all the nodes in the G-DAG. In addition, extra space (e.g., an array or hash table) for book keeping is needed to avoid repeated visits to the nodes. Since dominated sub-paths may be generated, a post-processing step is required to remove dominated sub-paths from the candidate list.

Simple graph traversal strategies such as BFS or DFS are not efficient on G-DAGs since dominated nodes in the G-DAG are visited. As a result, the candidate list contains

numerous dominated nodes, which requires an extra step to eliminate the dominated sub-paths. In order to traverse the G-DAG efficiently, one may want to visit as few nodes as possible. In order to evaluate the efficiency of different G-DAG traversal strategies, we propose the following definition.

Definition 8. ***Coverage:*** *The coverage of a G-DAG traversal strategy on a particular G-DAG is the ratio of nodes examined by the traversal strategy against the total number of nodes in the G-DAG.*

As can be seen from Figure 3.6(a), BFS always has a coverage of 100%, which is quite inefficient. By contrast, in the following part, we present two G-DAG traversal strategies adapted from our earlier work [7], namely, Bottom-Up traversal with in-Row Pruning (BURP) and Breadth-first traversal with Sub-Graph Pruning (BSGP). Then we present a new G-DAG traversal strategy, namely, Row-wise Traversal with Column-Pruning (RTCP). Figure 3.6 (b) - (d) illustrate these three strategies using the G-DAG of the sample path shown in Figure 3.1. Blue arrows represent the order of nodes examined by each strategy, after the leaf-evaluation (represented by black arrows). As can be seen, the total number of nodes examined by the BURP is 66-8 = 58. The coverage is thus 58/66 = 87.9%. The coverage of the BSGP and RTCP strategies are both 46/66=69.7%.

Preliminary Strategies for G-DAG Traversal

In our earlier work, our SEP approach employed a row-wise strategy and a top-down strategy. In this chapter we formulated them as G-DAG traversal strategies (after leaf-evaluation) and rename them as "Bottom-up traversal with in-row pruning (BURP)" and "Breadth-first traversal with sub-graph pruning (BSGP)".

Bottom-up traversal with in-row pruning (BURP): The corresponding G-DAG of the input path is traversed starting from the left-bottom leaf node. Each row in the G-DAG (corresponding to all the sub-paths ending at the same location) is examined from the left boundary node (longest sub-path) to the leaf (unit sub-path). Once a sub-path is identified as an ISP, the rest of this row will be skipped. Rows are examined from bottom to top. This traversal order follows the dominating relationships in each row. Figure 3.6(b) illustrates BURP algorithm after the leaf-evaluation on the

G-DAG of the sample data. Yellow nodes correspond to ISPs, and blue nodes correspond to the DISPs in the path. In this dataset, nodes (1, 2) and (5, 11) are DISPs (in blue color). When (5, 11) it is identified, the algorithm terminates the traversal on row 11 and moves to row 12. A refine step is then done to eliminate all the sub-paths that are dominated by others. Since at most one candidate ISP for each row (ending location) will be generated, the computational cost for the refine step is bounded to quadratic of the total number of unit sub-paths (the length of the input path). The pseudo code of BURP can be found in related work [7]. It is shown in Algorithm 1.

Table 3.2: Computing the number of parents

Case of node p	Number of Parents (m)
p is the root of the G-DAG	m = 0
p is on the boundary of the G-DAG	m = 1
p is a inner node of the G-DAG	m = 2

The BURP algorithm always follows the dominance relationship within each row in the G-DAG (longer sub-paths first). But it cannot guarantee an optimal traversal order across rows (sub-paths with different end locations). We thus propose a breadth-first traversal (after leaf-evaluation) with sub-graph pruning (BSGP) strategy which guarantees that a sub-path is always examined before all its subsets.

Breadth-first traversal with sub-graph pruning (BSGP): This traversal strategy completely follows the dominating relationship. As a result, the algorithm directly generate all the DISPs and the refine step is no longer needed. This strategy employs a Breadth-first traversal over the G-DAG, starting from the root node (longest ST sub-path). Figure 3.6(c) shows the traversal order of the BSGP algorithm after leaf-evaluation on the sample data. Yellow nodes are those corresponding to ISPs in the spatial path, and the blue ones (1,2) and (5,11) are the only DISP nodes discovered in the data. Black arrows and blue arrows represent the leaf-evaluation and BSGP traversal, respectively.

The traversal rule is as follows: In the traverse, a node will be evaluated only when both of its parents in the G-DAG have been evaluated but none were identified as ISP. The pseudo code of the BSGP traversal strategy after leaf-evaluation is shown in Algorithm 2. The algorithm uses a queue to do a breadth first search (BFS) with the

above constraint (line 3). For each node being evaluated, the algorithm computes all the distributive functions and the interest measure, and performs the test (lines 8-11) . If a node is found as an ISP node, both of its child nodes will be pruned. Otherwise, the algorithm will probe both of its children (lines 14-23). A key challenge is to determine if both parents of a node have been evaluated. Firstly, the number of parents of each node can be computed according to Table 3.2, which takes constant time. Secondly, a record table is built to keep track of how many more probes are expected. Every time a node is probed, the corresponding cell in the table is updated (lines 15-18). When the value of a node becomes zero, the node is pushed into the queue (lines 19-21). For example, if an inner node X is probed twice, we are sure that neither of its parent node is an ISP. Then we can safely evaluate this node. If one of its parents is identified as an ISP node, then node X will never get two probes, and will not be evaluated. Its successor nodes in the G-DAG will thus be pruned.

Proposed G-DAG Traversal Strategy

The previous subsection illustrated two preliminary G-DAG traversal strategies in our earlier work. A valid G-DAG traversal strategy may visit only a subset of the G-DAG but find all the DISPs. Although BSGP avoids unnecessary visits of dominated inner nodes, it uses a two-dimensional array to keep track of the visit record of each node in the G-DAG. This leads to two disadvantages: (1) high memory cost, and (2) extra time cost to handle the record table lookup and update. We could get some incremental optimization by using a hash table for the book-keeping to reduce the memory cost, However, the space complexity would still be $O(n^2)$ in the worst case, and the time cost won't improve.

In order to get rid of the extra space, we propose a smart traversal strategy, Row-wise Traversal (after leaf-evaluation) with Column Pruning (RTCP), which performs a revised row-wise traversal on the G-DAG. The pseudo code is shown in Algorithm 3. The traversal starts from the root node of the G-DAG, and goes to the right child of each node recursively, along each row. After reaching a DISP node or a leaf node, the algorithm stops the traversal in the current row, and switches to the next lower row, and starts the traversal again from the first column (i.e., the left boundary node). Note

that a global variable ($pBdr$) keeps the minimum column number of all the DISP nodes found so far among all the rows (Line 4). For example, when started from the root node, the $pBdr$ is 12 (the largest column number). When finding the DISP node (5, 11), the value of $pBdr$ changes to 5. The traversal along a row can terminate if no DISP node is identified before reaching this pruning border, since all the nodes beyond this point will be dominated by some existing DISP node (Lines 5-13). Figure 3.6(d) shows the traversal order of the RTCP algorithm. The blue arrows represent the traversal order, and the red line represents the location of the $pBdr$ after finding DISP node (5, 11).

As can be seen, the RTCP algorithm completely avoids redundant visits to dominated ISP nodes. It does not use extra space for the traversal. Comparison of the three traversal strategies: BURP, BSGP, and RTCP is given in Table 3.1.

3.4 Theoretical Analysis

This section presents a theoretical analysis of the proposed BURP, BSGP, and RTCP traversal strategies for G-DAG. We show proof of the correctness and completeness of the RTCP algorithm and the asymptotic complexity of all three algorithms. We also present a new cost model to evaluate the time cost of the algorithms.

3.4.1 Correctness and Completeness

Here we show proof of the correctness and completeness of the new RTCP algorithm. Correctness and completeness proofs of the BURP and BSGP algorithms are the same as shown in our earlier work [7].

Theorem 1. *The proposed bottom-up traversal with in-row pruning (BURP) algorithm is correct and complete. Correct here means that all the sub-paths discovered by the algorithm are DISP. Complete means that all the DISPs in the input dataset are reported by the algorithm.*

Proof. For the BURP algorithm, the identification of ISPs in the first phase follows directly from the definition. Any dominated sub-paths will be removed in phase 2. Thus the algorithm is correct. As for completeness, the algorithm only prune a node in G-DAG (sub-path) when (1) another ISP dominates this sub-path, or (2) the sub-path does not pass the interestingness test (and thus is not an ISP). Otherwise, this sub-path

will be added to the candidate set. In the second phase, again, only dominated sub-paths are removed from the result. So no DISP will be missing, and the BURP algorithm is complete. In summary, the BURP algorithm is both correct and complete. □

Theorem 2. *The proposed BSGP algorithm is correct and complete. Correct here means that all the sub-paths discovered by the algorithm are DISP. Complete means that all the DISPs in the input dataset are reported by the algorithm.*

Proof. In the BSGP algorithm, ISP identification follows straight from the definition. If a node p in the G-DAG is identified as an ISP, both of its successors will be pruned. Since the dominating relationship in the G-DAG has transitivity, no successor of p will be evaluated. This guarantees that no dominated ISP will be added to the candidate set. Thus, the BSGP algorithm is correct. The algorithm is complete if it does not miss any DISPs. As for the completeness, we prove that no DISP is missing. This is also guaranteed by the traversal rule: if a node p in the G-DAG is a DISP, none of the nodes dominating p is an ISP. In the algorithm, no pruning is done unless a node is identified as ISP. Thus no pruning will be performed at nodes dominating p, and node p will not be pruned by the algorithm. Thus, the BSGP algorithm is complete. As a summary, the BSGP is correct and complete. □

Theorem 3. *The RTCP algorithm is correct, i.e., all the nodes (i.e., sub-paths) identified by the RTCP algorithm are DISPs in the underlying ST path according to our definition.*

Proof. Apparently, each candidate node in the G-DAG (i.e., sub-path) found by the algorithm has been examined by the test of interestingness, and thus corresponds to an ISP in the underlying spatial path. According to the algorithm, if node (i, j) is an ISP, the $pBdr$ will be set to i so that no node in column k ($k \geq i$) will not be examined afterwards. Since the algorithm also examines rows in a descending order, all the nodes (i.e., sub-path) in row h ($h \leq j$) will be compared with the pruning border. This guarantees that no node (k, h) where $k \geq i,\ h \leq j$ (i.e., dominated by (i, j)) will be reported. So the proposed RTCP algorithm is correct. □

Theorem 4. *The RTCP algorithm is complete, i.e., all the DISPs in the underlying ST path are reported by the RTCP algorithm.*

Proof. In other words, a sub-path is only skipped when it is dominated by another ISP. The RTCP skips sub-path (i, j) only if the path starts in location $i \geq$ "*the current* $pBdr$". This means that there must have been an ISP (h, k) found where $h \leq pBdr \leq i$. Since at the time of skipping (i, j), sub-paths ending before j had not yet been examined, it can be derived that $k \geq j$. Thus (i, j) must be dominated by some ISP (h, k). This guarantees that a sub-path is only skipped when it is dominated by another ISP. Thus, the RTCP algorithm is thus complete. □

3.4.2 Complexity Analysis

Next we analyze and compare the time and space complexity of the three algorithms. As mentioned before, the analysis is only focused on the time and space complexity of the G-DAG traversal phase, since all three algorithms are designed based on the assumption that a lookup table is already built via leaf-evaluation. According to the property of the lookup table, computing any distributive function in any sub-path will take constant time. In the following analysis, we assume n is the number of unit sub-paths in the ST path S.

For BURP, in the first step, each row in the traversal space is examined from left to right. In the worst case, the patterns found lie at the end of each row. The time complexity of this step is $O(n^2)$. In the second phase, n candidates need to be examined and the time complexity is also $O(n^2)$. In the best case, however, all the DISPs lie along the left boundary of the G-DAG. The time complexity is thus $O(n)$. Since only one candidate is generated in each row, the time complexity of the second phase is also $O(n^2)$. In the extreme case where no interesting sub-path is found, the second phase is not needed but the total time complexity remains $O(n^2)$.

For the BSGP algorithm, the first and second steps are merged. In the best case, only one longest sub-path exists in the data which is the path S. Only constant time is needed to identify it. In the worst case, all the candidate sub-paths are at the bottom layer of the G-DAG. The entire traversal space needs to be examined and the time complexity is $O(n^2)$.

Now we look at the time complexity of RTCP. As shown previously, RTCP will examine only the necessary number of nodes in the corresponding G-DAG. This means

that in the best case, where the root node (entire path) is an ISP, the time cost of G-DAG traversal will be $O(1)$. In the worst case, the time complexity of G-DAG traversal is still $O(n^2)$ as the algorithm has to examine each individual sub-path to identify the shortest ISPs.

Regarding space complexity, the BURP algorithm takes $O(n)$ for the candidate list of the traversal after Step 1. The BSGP algorithm will take $O(n^2)$ for the book keeping of each node in the Breadth-First Search, which is also the total complexity. The RTCP uses only several variables (e.g., $pBdr$, c_Max) for auxiliary memory cost in the G-DAG traversal ($O(1)$).Detailed analysis results are shown in Table 3.3. As can be seen, the RTCP algorithm is always optimal in both time complexity and space complexity.

3.4.3 Cost model

The actual time cost of the G-DAG traversal strategies depends on a number of factors. Besides the total number of nodes evaluated, the number of visits to auxiliary data structures also affects the total time cost. The following cost model takes these factors into consideration to compare the time cost among the three algorithms. Assume there are n unit sub-paths in the input spatial path. Let W and R be the cost for a single write and a single read operation in the main memory, let P be the average number of parents in the G-DAG, m be the number of distributive functions to compute the interest measure, N be the total number of nodes in a G-DAG ($N = n \cdot (n+1)/2$) , and let C_1, C_2, and C_3 represent the coverage of the three strategies, respectively. The total time cost of each G-DAG traversal strategy thus can be expressed as follows:

$T_{BURP} = (2mR) \cdot C_1 \cdot N + n \cdot W + n \cdot (n-1) \cdot R$

$T_{BSGP} = (2mR + P \cdot (W + R)) \cdot C_2 \cdot N$

$T_{RTCP} = 2mR \cdot C_3 \cdot N$

The term $2mR$ represents the $2m$ read operations from the lookup tables to compute the score. In T_{BURP}, $n \cdot W$ is the cost for writing each candidate in the list, and $n \cdot (n-1) \cdot R$ is the cost for dominated sub-path elimination. In T_{BSGP}, $P \cdot (W + R)$ represents the $P - 1$ updates to the book-keeping array of each visited node and one write/read on the queue for BFS. In T_{RTCP} no extra cost is needed besides the first term. It is easy to derive from the proof to Theorem 3 and Theorem 2 that RTCP and BSGP

will never visit unnecessary node in the traversal and thus having the minimum coverage among all the G-DAG traversal strategies, i.e., $C_1 \geq C_2 = C_3$. So $T_{RTCP} \leq T_{BSGP}$ and $T_{RTCP} \leq T_{BURP}$. Though asymptotically the time complexity of the BSGP and RTCP strategies are the same with respect to the number of unit sub-paths, the actual time cost for RTCP is still lower than BSGP.

Table 3.3: Time and space complexity of the three algorithms

Algorithm	BURP	BSGP	RTCP
G-DAG traversal best case	$O(n)$	$O(1)$	$\mathbf{O(1)}$
G-DAG traversal worst case	$O(n^2)$	$O(n^2)$	$O(n^2)$
Dominated sub-path elimination	$O(n^2)$	NA	NA
Memory cost for G-DAG traversal	$O(n)$	$O(n^2)$	$\mathbf{O(1)}$

3.5 Experimental Evaluation

We evaluate the computational savings of the newly proposed RTCP algorithm over the previous BURP algorithm and BSGP algorithm. Specifically, we used a large synthetic dataset and a real dataset to evaluate the impact of input parameters on the performance of the RTCP algorithm and the two previous design decisions.

3.5.1 Experiment setup

In our experiment, we design a specific interest measure, namely, the sameness degree as an example to test our algorithm. This algebraic interest measure is chosen because it is suitable for finding change patterns in eco-climate data, which is of interest to our domain experts. Since we are not taking advantage of any specific assumptions besides assuming that the measure is "algebraic", we could easily generalize the experimental evaluation results to other algebraic interest measures.

Defining the unit sub-paths: We define the unit sub-path in the same way as shown in Figure 3.1. Given a ST path S with locations $s_1, s_2, ..., s_N$, and an attribute (e.g., vegetation cover) function f_v defined over it, the value associated with each unit sub-path is the difference between the values at two end locations, i.e.,

$diff(s_i, s_{i+1}) = f_v(i+1) - f_v(i)$.

Defining the Interest measure: Unit sub-paths with a significantly large difference are flagged as abrupt change unit sub-paths. Given an abruptness threshold θ_a, a unit sub-path u is an abrupt change unit sub-path if $diff(u) \geqslant \theta_a$. The sameness degree of a sub-path W (denoted as $SD(W)$) is defined as the ratio of two parts: the numerator is the average difference value of all the unit sub-paths in W, and the denominator is the average difference value of all the abrupt increase unit sub-paths. Formally, it can be written as: $SD(W) = \frac{AVG\{diff(w)\}}{AVG\{diff(w_a)\}}$, where w and w_a represent "all the unit sub-paths in W", and "all the abrupt change unit sub-paths in W", respectively. Specifically, if no abrupt change unit sub-path exists in W, $SD(W) = 0$.

This interest measure is an algebraic function of the $diff$ value of the unit sub-path in W, which can be decomposed into four distributive functions: $sum()$, $count()$, $sum\_if()$ and $count\_if()$. These functions are all reversible so that the lookup table design decision is enabled. We will illustrate the details of this measure and its domain interpretations in Section 3.6.

Defining the test and pattern: A sub-path W is interesting if its sameness degree $SD(W) \geqslant \theta_{sd}$, where θ_{sd} is a user specified threshold between 0 and 1.

We varied the total number of unit sub-paths in the traversal space (n) to test the scalability of the proposed algorithms. Also, we still use the proposed parameter, pattern length ratio (PLR), to test the algorithms' sensitivity to the pattern length.

Pattern Length Ratio (PLR): As discussed previously, the time cost of the algorithms partially depend on the length of the DISPs in the input data (best and worst cases). Intuitively, loosening the interestingness test (the sameness degree threshold θ_{sd}) will lead to longer patterns. However, the actual pattern length is hard to control using θ_{sd}. Instead, we directly manipulate the data distribution to control the pattern length. The maximum length of the DISP divided by the total number of unit sub-paths, is defined as the pattern length ratio. PLR = 1 will give one longest pattern in the data, and the minimum PLR means that all the patterns have only the minimum length.

Datasets: We used a synthetic dataset and a real dataset in the experiments. The synthetic data were randomly generated with 50,000 unit sub-paths. We manipulate the data in such a way that the PLR of the dataset can be controlled. We use a real time series from climate science to evaluate the performance of our algorithms. The

time series is a global averaged daily maximum temperature generated by the GFDL coupled model [128]. The dataset covers a range from Jan. 1, 1861 to Dec. 31, 2000, with 51100 data values. We converted it to a temporal path with 51099 unit sub-paths

We implemented the algorithms and the testing program in C++. All experiments were performed on Intel Xeon 2.53GHz 4 Core Workstation with 11.8GB RAM, under Ubuntu Linux 10.04 system.

3.5.2 Question 1: How does pattern length ratio (PLR) affect the time cost for the three algorithms?

We first test the performance with respect to pattern length. We fixed the number of unit sub-paths in each path at 20,000 (number of leaf nodes in the G-DAG). We aligned the longest pattern with the beginning of the path. The PLR was varied from 0.1 to 1.0, corresponding to the worst case for all the algorithms, and the best case for the BSGP and RTCP algorithms. Figure 3.7 shows the run time of the three algorithms. The RTCP algorithm is much faster than the BURP and BSGP algorithms in all cases. Only in the best case (PLR = 1.0) does BSGP finally achieve the same run time as the RTCP.

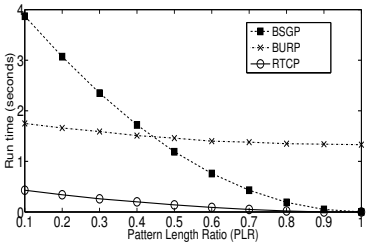


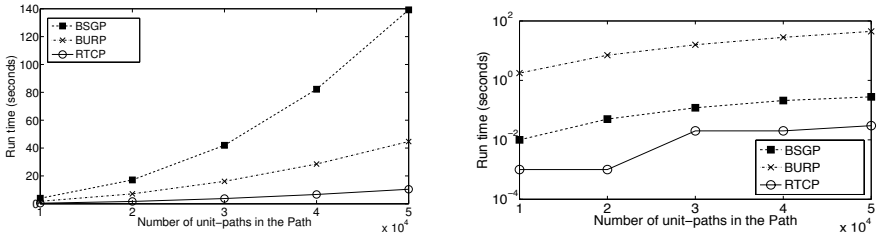
Figure 3.7: Run time of the three algorithms on synthetic datasets

3.5.3 Question 2: How does increasing path length affect the time cost of the algorithms?

We then test the scalability of the algorithms on synthetic datasets. We fixed the PLR at 0.1 (corresponding to the worst case) and varied the total number of unit sub-paths from 10,000 to 50,000. Figure 3.8(a) shows the run time of the three traversal strategies in this scenario. The RTCP algorithm is always orders of magnitude faster than the BURP

and the BSGP algorithms wIth speedups over the BURP and the BSGP algorithms of 76% and 90%, respectively. This is due to the fact that the RTCP algorithm traverses the G-DAG more efficiently and does not need to maintain extra memory and data structure. We then fixed the PLR at 1.0 (corresponding to the best case) and repeated the test. Figure 3.8(b) shows that the RTCP algorithm is also much faster than the two other algorithms in this scenario. Note that for the RTCP algorithms, the run-times for $n = 20000$ is still very close to zero so that the plot shows a flat trend.

We also ran our algorithm on the real dataset. We manipulate the abruptness threshold and sameness degree threshold to make the worst (PLR = 0.1) and best (PLR = 1) cases. We varied the total number of unit sub-paths used in the experiments from 10,000 to 50,000. For the worst case (Figure 3.9(a)), the RTCP algorithm always runs faster than both BSGP (70% speedup) and BURP (25% speedup). For the best case (Figure 3.9(b)), RTCP and BSGP algorithms have the same performance, which is orders of magnitude faster than the BURP algorithm.



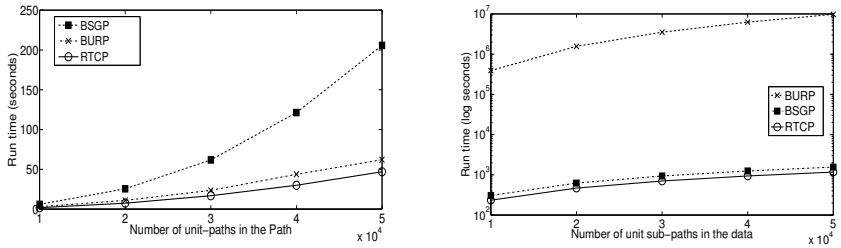
(a) Time cost with PLR=0.1 and varying path length on synthetic dataset

(b) Time cost with PRL=1.0 and varying path length on synthetic dataset (log scale)

Figure 3.8: Run time of the three algorithms on synthetic datasets.

3.5.4 Question 3: How do the total memory costs of the three algorithms compare?

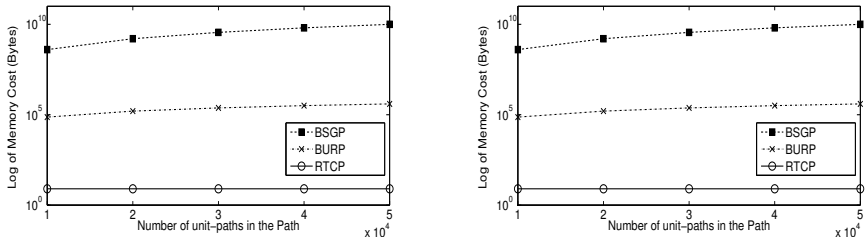
We finally compare the total memory cost of the three algorithms with an increasing number of total unit sub-paths (data volume). In the experiments, we count the total size of memory (bytes) used by the three G-DAG traversal strategies, including auxiliary data structures (e.g., the queue for Breadth-first traversal and the book-keeping array) and intermediate results. We do not count the memory cost of the lookup tables built in the leaf-evaluation phase. Figure 3.10(a), shows the total memory cost (bytes) in log



(a) Time cost with PLR = 0.1 on real dataset

(b) Time cost with PRL= 1.0 on real dataset (log scale)

Figure 3.9: Run time of the three algorithms on the real dataset.



(a) Results on synthetic data

(b) Results on Real data

Figure 3.10: Memory usage of the three G-DAG traversal strategies (in log scale).

scale on the synthetic dataset. The memory cost of RTCP remains constant, which is orders of magnitude less than the memory cost of BURP and BSGP. The memory cost of BURP and BSGP increase at linear and quadratic speed, respectively. The results on the real dataset (shown in Figure 3.10(b)) are very close to the result on the synthetic data, since the memory cost of the algorithms only depend on the total number of unit sub-paths.

To summarize, the proposed RTCP algorithm is always faster than the BURP algorithm with any pattern length. It is also always much faster than the BSGP algorithms except in the case when the entire path is interesting and only one evaluation is needed. The new algorithm is also more scalable and significantly faster than our previous algorithms with growing data lengths. Finally, RTCP always has a lower memory cost than the other two algorithms. Thus RTCP dominates the competitors in both time and space cost.

3.6 Case Study on Ecoclimate Data

We applied the proposed RCTP approach on eco-climate data used in climate change research. The goal was to show that the algorithm can discover useful ST sub-paths. In the rest of this section, we use the sameness degree introduced in the previous section to discover sub-paths of abrupt change. Finally, we show the results of the approach with interpretation by domain scientists.

3.6.1 Discovering ST Sub-paths of Abrupt Change

Climate change researchers are interested in patterns of change in eco-climate data. As noted earlier, areas(sub-paths) in a geographical space displaying evidence of abrupt changes in rainfall, vegetation cover, etc. may signal the presence of ecotones between different ecological zones. In the temporal dimension, sub-paths may reflect climatic shifts occurring rapidly over time. In computational terms, given a ST path in eco-climate data, the goal is to discover all the sub-paths along which there is a consistently abrupt change in one or more attributes. We define the interest measure for the pattern we are seeking as the "degree of sameness." As has been illustrated in the previous section, the sameness degree of a sub-path W (denoted as $SD(W)$) is formally defined as: $SD(W) = \frac{AVG\{diff(w)\}}{AVG\{diff(w_a)\}}$, where w_a represents all the abrupt change unit sub-paths in W. Specifically, if no abrupt change unit sub-path exist in W, $SD(W) = 0$. One way to specify the abruptness threshold θ_a is to compute certain quantile (e.g., top 10%) of the population of all the unit difference values.

The sameness degree measures the "slope" of values in a sub-path against its abrupt part, thereby showing the "sameness" of the increasing trend in sub-path. Due to the property of the sameness degree, its value is always between 0 and 1. A larger value means a more interesting pattern. A sameness degree of 1 means that all the unit sub-paths are abrupt change unit sub-paths, while a sameness degree of 0 means that the change is not an abrupt increase at all. The sameness degree is better than simple slope in that it is bounded and less sensitive to noise data.

A sub-path W is an interesting sub-path if it has a sameness degree $SD(W) \geqslant \theta_{sd}$, where θ_{sd} is a user specified threshold between 0 and 1. We call such sub-paths "Sub-paths of abrupt change". For simplicity, we limit the search space by adding a constraint that the start and end unit sub-paths of an interesting sub-path are abrupt change unit

sub-paths. Finally we define the "dominant sub-path of abrupt change" as a "sub-path of abrupt change" that is not a subset of any other "sub-path of abrupt change".

3.6.2 Datasets and Settings

In the case study, we used three datasets. The first was Normalized Difference of Vegetation Index (NDVI) data, measuring the extend of vegetation cover in Africa from Global Inventory Modeling and Mapping Studies (GIMMS) [32, 33]. The spatial resolution was 0.07 degree and the time period was from August, 1981 to 2006, with a snapshot every two weeks. We used one snapshot and smoothed the data using a longitudinal moving average of neighboring pixels in one degree.

The second dataset was Top Soil Layer Soil Moisture data from the MERRA Land project [129]. The value are measured in m^3 water per m^3 soil. The data resolution is 0.5 degree longitude and 2/3 degree latitude. We use the data from August, 1981 in Africa to verify our approach.

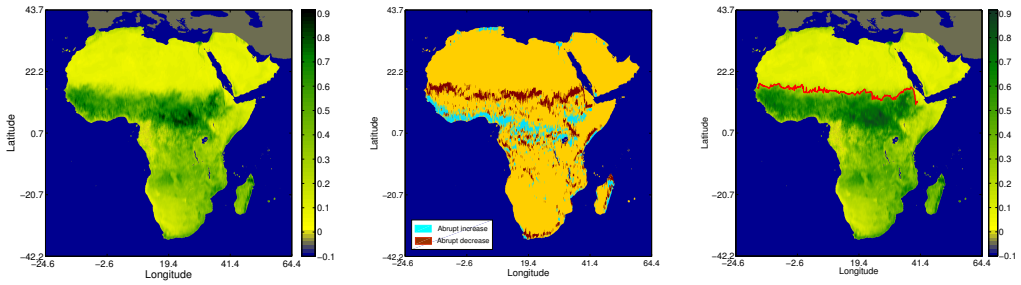
The third dataset was a rainfall index [31, 130], specifically a time series of summer (June to October) precipitation in the Sahel area, Africa, from 1900 to 2010. The data was normalized with respect to the mean from 1900 to 2010. We also smoothed the data, using a five-year moving average.

3.6.3 Discovery of Ecotones

We use both the vegetation and soil moisture data to discover the footprint of ecotones (e.g., Sahel). The both of the two datasets contained multiple spatial paths. For simplicity andbetter illustration of the pattern, we chose only spatial pathsalong each longitudinal column in the dataset, from south to north.We ran the proposed RTCP algorithm to discover both sub-paths of abruptincrease and sub-paths of abrupt decrease.

Figure 3.11 shows the result of the RTCP algorithmon the Africa NDVI data, August 1-15, 1981. The dimension of the mapwas 1152 by 1152 pixels. We set the abruptness threshold $\theta_a$to the top 10% quantile of all the positive unit sub-path $diff$ values, and the samenessscore to 0.5. As indicated by the red and blue are as in the figure,RTCP discovered several ecotones in Africa. One of them is the wellknown Sahel region(in the middle in red), where vegetation coverexhibits an abrupt decreasing trend from south to north. The lightblue region in the middle is the southern boundary of the tropicalrain

forest.



(a) Smoothed Africa vegetation dataset (measured in NDVI) in August, 1981.

(b) Longitudinal sub-paths of abrupt vegetation change in August, 1981

(c) Change points on each longitudinal path discovered by CUSUM

Figure 3.11: Sub-paths of abrupt vegetation changes discovered over Africa along longitudes (best viewed in color).

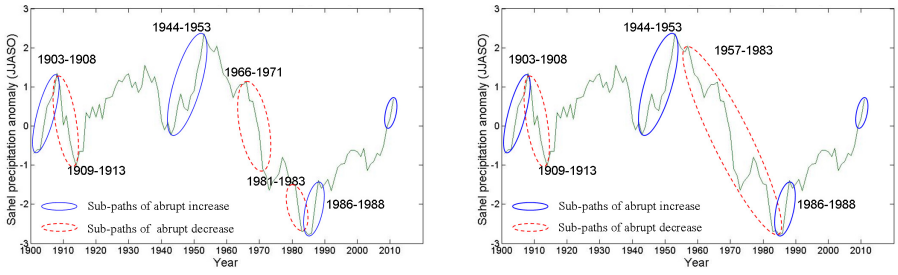
In order to compare the result of RTCP with related work, we also ran a variation of the CUSUM [70] algorithm on the same dataset. Each longitudinal path is processed by the offline CUSUM algorithm to find a single change point where the NDVI value changed significantly compared with the mean NDVI of the entire longitudinal path. The change points across different longitudes form a line on the map (the red line shown in Figure 3.11(c)). Apparently, the CUSUM algorithm could not discover the full extent of the ectones as it only finds a thin line. This is less informative to domain experts when analyzing the interactions between ecotones and climate change. The proposed approach is more effective and able to address the limitation of related work.

3.6.4 Discovery of Abrupt Precipitation Shifts

We also applied the RTCP algorithm on the smoothed Sahel rainfall index time series to discover temporal sub-paths of abrupt precipitation change in this region. RTCP was used to analyze the smoothed Sahel precipitation anomaly data, shown in Figure 12. By setting the sameness degree threshold to 0.5 and the abruptness threshold to the top 25% quantile of all the positive unit sub-path $diff$ values, we discovered a few major sub-paths of abrupt precipitation change (as shown in Figure 3.12(a)). Abrupt increases

and decreases are highlighted using blue and red ellipse respectively. For example, the period from 1967 to 1971 shows the well known abrupt decline of precipitation for those years in the Sahel [79].

We also identified several abrupt increases (1903-1908, 1944-1953, 1986-1988 and 2008-2010), which have been discussed in literature [131, 132]. For example, rainfall in Sahel seems to have started recovering after 1986. By reducing the sameness degree threshold to 0.3, we discovered a longer sub-path of abrupt precipitation decrease from 1957 to 1983 which includes the two shorter sub-paths (1968-1971 and 1981-1983) we previously discovered, shown in Figure 3.12(b). This decreasing trend has been mentioned in the literature [80]. Overall, our case study shows that the proposed approach can indeed discover useful patterns from real datasets.



(a) Result for $\alpha = 0.25$ and $\theta_{sd} = 0.5$ (b) Result for $\alpha = 0.25$ and $\theta_{sd} = 0.3$

Figure 3.12: Temporal sub-paths of abrupt rainfall increase and decrease in the Sahel region (best viewed in color).

3.6.5 Parameter Selection and Interest Measure Generalization

The selection of proper thresholds in the above case studies was guided by domain experts with interactive visual analysis. As one of the major contribution, our efficient computational framework enables timely result feedback for interactive data exploration and parameter setting by domain users. Even though the selection of these specific parameters might be non-trivial for non-expert users, this approach also has its value since it can be applied in other domains. Users in any domain can define algebraic interest measures with their own parameters and threshold selection accounting for various statistical features of sub-paths and still use the proposed approach for efficient computation.

3.7 Discussion

This section discusses a broader spectrum of other problems dealing with ST paths. These problem, however, are not directly related to our work as the objectives of these studies are quite different from the problem in this work. A specific ST sub-path discovery problem, namely flow anomaly detection, aims to find interesting temporal sub-paths in sensor readings where a high ratio of flow anomaly exists [133]. The approach defines a specific interest measure (persistence ratio), and deals with only Boolean data values(anomaly exits or not). The problem of trajectory outlier detection aim to find outlying segments in a trajectory with respect to other trajectories in a trajectory database [134]. The sequence/subsequence matching problems, such as longest common subsequence [135, 136], dynamic time wrapping [137, 138], similarity search [139], and trajectory clustering [140] focus on finding subsequences that match with a given query sequence or a group of similar subsequences. These problems are quite different from ours.

3.8 Summary and Future work

This chapter investigates the problem of discovering interesting sub-paths from a spatiotemporal path. In our recent work, we proposed a novel Sub-path Enumeration and Pruning (SEP) approach, with two computational designs for the problem. In this chapter, we model the problem as a Grid-based Directed Acyclic Graph (G-DAG) traversal problem, and present a new traversal algorithm, namely, the Row-wise traversal (after leaf-evaluation) with Column Pruning (RTCP) algorithm. Theoretical and experimental evaluation results show that the proposed RTCP algorithm is complete, correct, always runs much faster than the previous two design decisions with various pattern lengths, and achieves better scalability on both synthetic and real datasets. More importantly, RTCP has orders of magnitude less memory cost compared to the previous G-DAG traversal strategies in SEP.

In addition to improving algorithm performance, our future plans include further investigation of new interest measures for better modeling of the patterns, with parameters that are easier to choose. We also would like to add statistical significance testing to the computational framework to remove patterns generated by random chance. Finally,

we plan to apply the proposed approach to other types of interesting sub-paths (e.g., long sub-paths with extreme value) or other application datasets (e.g., traffic data).

Algorithm 1 Bottom-up traversal with in-row pruning (BURP) strategy after leaf-evaluation

Require:

- A path S with n unit sub-paths
- Values associated with unit sub-paths
- An interest measure (algebraic aggregate function) F_{isp}
- An interestingness test T.
- A lookup table built via leaf-evaluation

Ensure:

- All the nodes corresponding to DISPs in S.

1: $G \leftarrow$ the G-DAG representation of S
   //Step 1: ISP identification
2: $CanSet \leftarrow \varnothing$
3: **for** $n\_row \leftarrow 2$ to N **do**
4: **for** $n\_col \leftarrow 1$ to $n\_row - 1$ **do**
5: Compute distributive functions D^1_{aggr}, D^2_{aggr}, ..., D^m_{aggr} for node (n_col, n_row)
6: Compute F_{spi} using $D^1_{aggr}, D^2_{aggr}, ..., D^m_{aggr}$
7: **if** $T(F_{spi})$==True **then**
8: Add node (n_col, n_row) to $CanSet[]$
9: **Break**
10: **end if**
11: **end for**
12: **end for**
13:
   //Step 2: dominated ISP elimination
14: **for** $i = 1$ to $size(CanSet)$ **do**
15: **for** $j = 1$ to $size(CanSet)$ **do**
16: **if** $i \neq j$ and $CanSet[j] \subset CanSet[i]$ **then**
17: Remove $CanSet[j]$ from CanSet
18: **end if**
19: **end for**
20: **end for**
21: $Output \leftarrow CanSet[]$

Algorithm 2 Breadth-first traversal with Sub-Graph Pruning (BSGP) after leaf-evaluation

Require:

- A path S with n unit sub-paths
- Values associated with unit sub-paths
- An interest measure (algebraic aggregate function) F_{isp},
- An interestingness test T.
- A lookup table built via leaf-evaluation

Ensure:

- All the Dominant Interesting Sub-paths (DISP) in S.

1: $G \leftarrow$ the G-DAG representation of S
//Step 1: ISP identification
2: $CanSet[] \leftarrow \varnothing$, $ptv[] \leftarrow \varnothing$
3: Create an empty queue Q
4: Q.enqueue($G.root$)
5: **while** Q is not empty **do**
6: $W \leftarrow Q$.head()
7: Q.dequeue()
8: Compute distributive functions D^1_{aggr}, D^2_{aggr}, ..., D^m_{aggr} for node W
9: Compute F_{spi} using $D^1_{aggr}, D^2_{aggr}, ..., D^m_{aggr}$
10: **if** $T(F_{spi})$==True **then**
11: Output(W)
12: **Next Loop**
13: **end if**
14: **for** each child node W_s of W **do**
15: **if** W_s is probed for the first time **then**
16: $ptv[W_s] \leftarrow$ number of W_s's prarents -1
17: **else**
18: $ptv[W_s] \leftarrow ptv[W_s] - 1$
19: **if** $ptv[W_s] == 0$ **then**
20: Q.enqueue(W_s)
21: **end if**
22: **end if**
23: **end for**
24: **end while**
//Step 2: Not needed

Algorithm 3 Row-wise Traversal with Column Pruning (RTCP) after leaf-evaluation

Require:

- A path S with n unit sub-paths
- Values associated with unit sub-paths
- An interest measure (algebraic aggregate function) F_{isp}
- An interestingness test T.
- A lookup table built via leaf-evaluation

Ensure:

- All the nodes corresponding to DISPs in S.

1: $G \leftarrow$ the G-DAG representation of S
//Step 1: ISP identification
2: $pBdr = N$
3: **for** $n\_row \leftarrow N$ to 2 **do**
4: $c\_Max = MIN(pBdr, n\_row - 1)$
5: **for** $n\_col \leftarrow 1$ to c_Max **do**
6: Compute distributive functions D^1_{aggr}, D^2_{aggr} ,..., D^m_{aggr} for node $W = (n\_col, n\_row)$
7: Compute F_{spi} using $D^1_{aggr}, D^2_{aggr}, ..., D^m_{aggr}$
8: **if** $T(F_{spi})$==True **then**
9: Output(W)
10: $pBdr = n\_col$
11: **Break**
12: **end if**
13: **end for**
14: **end for**
//Step 2: Not needed

Chapter 4

Discovering Persistent Change Windows in Spatiotemporal Datasets

4.1 Introduction

Given a region S comprised of locations that each have a time series of length $|T|$, and a change rate threshold, the problem of Persistent Change Window (PCW) discovery aims to find all (window, interval) pairs $\langle S_i, T_i \rangle$ that exhibit persistent change over time. For example, Figure 4.2 shows a sample spatiotemporal data field with 16 locations. Each location is associated with a series of 4 values indicating the vegetation cover at each time step. Given a minimum average change rate across time steps, the PCW discovery may find a rectangular window (e.g., the nine locations to the left-top corner) and a time interval (e.g., time steps 1 through 4) where a persistent degradation of the land cover occurs.

PCW discovery is important to a number of societal applications. Ecologists, for example, may be interested in identifying regions where the landscape progresses through different stages and continuously transforms in appearance. Urban planners and policy makers may be interested in finding regions where the farmland grows rapidly to

assess urban sprawl and food production. Climate scientists may be interested in finding regions where a persistent decrease in precipitation occurs to assess the severity of droughts. The explosion of planetary environmental data in recent years created new opportunities for answering the above questions. For example, Google's Earth Engine is comprised of trillions of scientific measurements dating back almost 40 years [1]. Figure 4.1 shows an example in Brazil from 1984 to 2012 where the effect of deforestation of the Amazon becomes more pronounced over time. Identifying geographic areas that may be exhibiting persistent change patterns, however, is a labor intensive task for domain experts dependent on visual analysis of the data. Efficient computational approaches to help in early identification of such regions may facilitate countermeasures or prevention techniques such as reforestation, where depleted forests and woodlands may be restocked.

(a) 1984 (b) 1998 (c) 2012

Figure 4.1: Amazon Deforestation in Brazil (Courtesy: Google Earth Engine [1]) (Best in color).

Increasingly, however, the size, variety, update rate, and combinatorics (e.g., the enumeration space of candidate patterns) of spatial datasets such as Google Earth Engine [1] exceed the capacity of commonly used spatial computing and database technologies to learn, manage, and process the data with reasonable effort. We believe that this data, which we call *Spatial Big Data (SBD)*, represents the next frontier in several domains including Climate Science, Ecology, and Urban Planning. In addition to data from Google's Earth Engine [1], examples of emerging SBD datasets include Unmanned Aerial Vehicle (UAV) data, LiDAR data, etc.

PCW discovery is challenging for the following reasons. First, there are a huge number of candidate patterns to consider when trying to determine the solution. For

example, if the given spatial window S is comprised of $M \times N$ locations, where each location has a time-series of length $|T|$, the number of candidates is $M^2 \times N^2 \times T^2$. For a moderate resolution remote sensing data tile (e.g., MODIS 250 NDVI, 4800 by 4800 pixels for 13 years), the total number of candidate may reach 10^{16}. If we consider all the tiles in the dataset or finer resolution dataset (e.g., Landsat 30m resolution) at the global scale, the candidate space will exceed 10^{30}, i.e., "big combinatorics". Second, PCW lacks monotonicity; regions of interest may be comprised of sub-regions that are not interesting, making computational techniques such as apriori-based pruning and dynamic programming inapplicable. For example, in Figure 4.1, there are several regions without deforestation inside and around the regions with deforestation, which illustrates the lack of monotonicity. Third, the size of an ST window may vary, without a maximum length. For example, deforestation in Brazil has spanned over 230,000 square miles since 1970 [141]. Finally, the data volume is potentially large when considering attributes such as vegetation cover, temperature, precipitation, etc., over hundreds of years from different global climate models and sensor networks. The volume of such datasets will range from terabytes to petabytes.

Previous approaches on Spatiotemporal (ST) change footprint discovery has focused on discovering abrupt change point with local (e.g., time series at individual raster cells or pixels)or zonal footprints. Local-footprint based methods include time series change point detection [122, 69, 123, 18] techniques, which aim at finding a time point in a single time series when a shift in statistical parameter occurs. For example, CUSUM [69, 123] keeps a cumulative score of the log-likelihood ratio of distribution parameter (e.g., mean), and flags the change when the score exceeds a threshold. Zonal techniques such as spatiotemporal scan statistics [109, 107] takes the aggregated attributes series (e.g., total number of disease count) in each spatial area, and finds an area and time point where a change in the underlying distribution is most likely to occurs (e.g., outbreak). These related techniques may find the most likely change pattern, but do not guarantee completeness of the results. In addition, they may not directly solve the problem but only provide assistance to manual efforts.

In addition to the above work, a large body of change detection techniques [142, 12, 143] and softwares [144, 145] developed in remote sensing research have focused on finding pixel-wise changes across a few (typically two) snapshots of satellite images.

These techniques, though having the ability to find changes across multiple snapshots, lack the ability to discover time intervals with arbitrary length and non-monotonic changes. In addition, these techniques output pixel-wise changes rather than zonal, collective summarization of change footprints, and may require intensive human labor for post-processing and visual analysis when dealing with big data.

Other work expands the temporal change footprint to intervals or periods of interest [7]. However, these work are still local- footprint based, and have not been extended to handle persistent change patterns with zonal spatial footprint. In contrast, in this work we propose a completely automatic and complete computational approach to discover persistent change patterns with a zonal spatial footprints. Table 4.1 summarizes the classification of related work in this area.

Table 4.1: Classification of Related Work

Spatiotemporal (ST) Change		Spatial footprint	
		Local	Zonal
Temporal footprint	Point (transient)	CUSUM [122, 69, 123]	ST scan statistic [109]
	Across few snapshots	Remote sensing change detection [142, 12, 143]	
	Long Interval (persistent)	Interesting subpath discovery[7]	**Our Work**

Contributions: To address the above limitations of related work, we propose a space-time (ST) window enumeration and pruning (SWEP) approach that considers zonal spatial footprints when finding ST windows. It is completely automatic and guarantees the completeness of results. In summary, our contributions are as follows:

- We formally define the persistent change window (PCW) discovery problem.
- We propose an ST window enumeration and pruning approach (SWEP) as a computational solution to PCW discovery problem
- We provide theoretical analysis of the correctness, completeness, and space/time complexity of the proposed method

- Experiments on a synthetic datasets with various settings show that SWEP leads to orders of magnitude computational savings over the naive approach
- We present a case study using vegetation cover data to evaluate the effectiveness of SWEP.

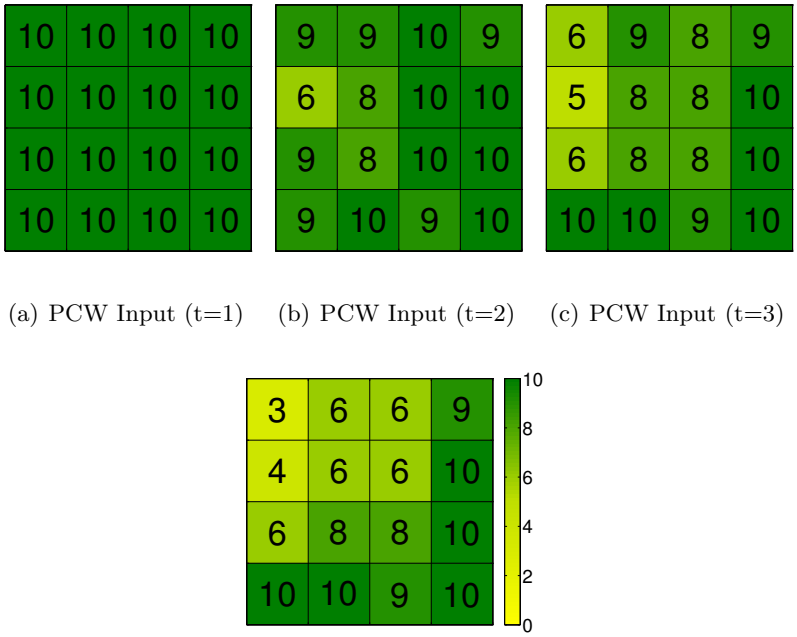
Scope and Outline: This chapter is focused on ST windows with persistent changes. Visual analytics and other semi-automatic or manual techniques for land cover change analysis are beyond the scope of this chapter. The proposed approach is validated using case study of vegetation cover change. However, we do not discuss the details (e.g., causes or impacts) of desertification, deforestation, etc. The rest of this chapter is organized as follows: Section 4.2 presents the basic concepts and problem statement of PCW. Section 4.3 outlines a naive approach to solving PCW and presents our proposed SWEP algorithm. In Section 4.4, we provide theoretical analysis of the correctness, completeness, and space/time complexity of SWEP. Section 4.5 presents a case study that shows the effectiveness of SWEP using vegetation cover data. Section 4.6 outlines the experimental evaluation. Section 4.7 concludes the chapter and discusses future work.

4.2 Basic Concepts and Problem Statement

In this section, we introduce several key concepts in the Windows of Persistent Change (PCW) discovery problem and give a formal problem statement.

4.2.1 Basic Concepts

Relevant definitions to our problem statement and proposed approaches are as follows: **Spatial Time-Series:** A *spatial field,* S, is a partition of a region of geographic space, forming a finite tessellation of spatial objects or locations. An example of a spatial field is shown in Figure 4.2. As can be seen, the spatial field is comprised of 16 locations, each having different values at different time instants. A *temporal framework,* T, is a partition of a time interval into sub-intervals and a *time-series* is a computable function from T to a finite attribute domain, A_i. A *spatial time-series* is $S \times T$, which may be



(a) PCW Input (t=1) (b) PCW Input (t=2) (c) PCW Input (t=3)

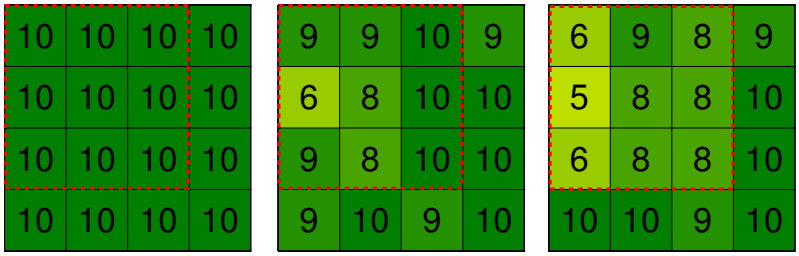
(d) PCW Input (t=4)

Figure 4.2: Example input of Persistent Change Window (PCW) discovery(Best in color).

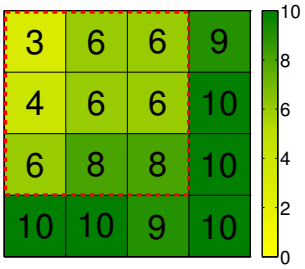
thought of as a spatial field where each location $s_i \in S$ has a time-series. In Figure 4.2, the time-series for the spatial location in the upper left corner is $[10, 9, 6, 3]$, indicating different values at that location (e.g., vegetation cover) at time instants 1 through 4.

Spatial and spatiotemporal window: A spatial window S_i in a spatial field S with $M \times N$ locations is defined as $S_i = [x_1, x_2] \times [y_1, y_2]$, where $[x_1, x_2] \subseteq [1, M], [y_1, y_2] \subseteq [1, n]$, i.e., rectangular areas. Similarly, a spatiotemporal (ST) window is a subspace $S_i \times T_i$ where S_i is a spatial window and T_i is a time interval: $T_i = [t_1, t_n] \subseteq T$.

Spatial Aggregated Time-Series, T_{S_i}: Each location s_j in a spatial field S has a time-series $T_{s_j} = [x(s_j, 1), x(s_j, 2) \ , .., \ x(s_j, t)]$. A *spatial aggregated time-series* T_{S_i} over spatial window S_i is a series of aggregated value of the locations in S_i. For example, T_{S_i} can be defined as the sum (or average, etc) of time series in each location, i.e., $T_{S_i} = [\sum_{s_j \in S_i} x(s_j, 1), \sum_{s_j \in S_i} x(s_j, 2) \ ,..., \sum_{s_j \in S_i} x(s_j, t)]$. The spatial aggregated time-series for all 16 locations in Figure 4.2 is $[160, 146, 134, 121]$, where the value for each time instant t_i



(a) PCW Output (t=1) (b) PCW Output (t=2) (c) PCW Output (t=3)



(d) PCW Output (t=4)

Figure 4.3: Example output of Persistent Change Window (PCW) discovery (Best in color).

represents the sum of all the values of all the locations in t_i. For example, the value for time instant 1 is 160 because all 16 locations in the spatial field have a value of 10.

Average Change Rate (ACR): For a spatiotemporal window $S_i \times T_i$ (where $T_i = [t_1, t_n]$), the average change rate is defined as the average percentage of change (i.e., decrease or increase) in the spatial aggregated time series T_{S_i} over period T_i. Formally, it can be expressed as $ACR(S_i, T_i) = [T_{S_i}(t_1) - T_{S_i}(t_n)]/T_{S_i}(t_1)/(n-1)$, where $T_{S_i}(t_1)$ and $T_{S_i}(t_n)$ are the first and nth value of the spatial aggregated time series of window S_i. In Figure 4.2, the change (decrease) rate between time instants t_1 and t_2 for all 16 locations is 8.8%. The ACR in Figure 4.2 for all 16 locations and all 4 time instants is 8.13%, which represents the decrease rate between the first and last value, divided by the length: (160-121)/160/3 = 8.13%. An example of how the average change rate may be used is to determine the average rate at which total vegetation cover in an area (e.g., the Amazon) decreases over a certain period (e.g., the last 3 decades). Similarly, it can

be used to determine the average increase rate.

Persistent Change Window (PCW): Given a threshold r of minimum average change rate (ACR), a persistent change window is a spatiotemporal window $S_i \times T_i$ where $ACR(S_i, T_i) \geq r$.

4.2.2 Problem Statement

The problem of windows of persistent change (PCW) can be expressed as follows:
Given:

- A spatial field S with $M \times N$ locations that has a time series of values of length $|T|$
- A threshold r of average change rate (ACR)
- A minimum window size S_{min} (optional)
- A minimum time length t_{min} (optional)

Find:

- All persistent change windows $\langle S_i, T_i \rangle$

Objective:

- Reduce computational cost

Constraints:

- $|S_i| \geq S_{min}$ and $|T_i| \geq T_{min}$
- $\langle S_i, T_i \rangle$ is not a subset of any other pair $\langle S_j, T_j \rangle$ such that $S_i \subseteq S_j \wedge T_i \subseteq T_j$
- Completeness and Correctness

The inputs for PCW discovery include a spatial field, an average change rate threshold, and minimum window and time length sizes as defined previously. The output is all window-interval pairs whose ACR exceeds the given threshold. The objective is to reduce computational cost. The problem has three constraints. First, window and time intervals require user-specified minimum sizes, which allows flexibility in ignoring small window-interval pairs which may not be of interest. A second constraint is that $\langle S_i, T_i \rangle$ pairs must not be subset of other window-interval pairs. This avoids duplication in that

the information in a smaller $\langle S_j, T_j \rangle$ pair that is a subset of $\langle S_i, T_i \rangle$ is fully captured by the larger or dominant pair. The third constraint is completeness (i.e., all relevant $\langle S_i, T_i \rangle$ pairs are discovered) and correctness (i.e., all discovered $\langle S_i, T_i \rangle$ pairs are indeed persistent change windows as outlined in the problem statement).

Example: Figure 4.2 and Figure 4.3 show input and output examples of PCW. The input, shown in Figure 4.2, consists of a spatial field comprised of 16 locations, where each location has a time series of 4 values. The ACR threshold is set to 15%, and S_{min} and T_{min} are set to 9 and 4, respectively. The output, shown in Figure 4.3, is the highlighted window (9 locations) across all 4 time instants. The highlighted $\langle S_j, T_j \rangle$ pair has an ACR of 16%, which satisfies the threshold. The ACR is based on the spatial aggregated time-series for the highlighted 9 locations in the spatial field, which is [90, 79, 66, 53]. The total change rate over this time period is (90-53)/90=52%. The ACR is the average of the decrease rate over 3 years, i.e., 52%/3 = 17.3%).

An important difference between the proposed problem and previous work in the vast domain of remote sensing [142, 12, 143] is that in PCW all region and time-interval pairs are considered. This is very different from calculating persistent changes on a pixel-by-pixel basis or calculating persistent changes for a pre-defined region. If the work of looking at every region and time-interval pair is not done, important windows across space and time may be missed. For example, in Figure 4.3, the ACR between time instants t_1 and t_2 for the pixel in the first row, second column is 10% whereas the ACR for the pixel in the first row, third column is 0% for the same period. Individually, only the former pixel might seem interesting. However, PCW allows us to analyze these pixels (and every other combination of pixels) as a group in space and time. This advantage provides early identification of regions of deforestation, urban sprawl, etc.

4.3 Proposed Approach

This section describes the naive algorithm and our proposed ST window enumeration and pruning (SWEP) approach for solving the ST windows of persistent change (PCW) discovery problem.

4.3.1 Naive Approach

We first present an intuitive and brute-force solution as the baseline. The pseudo code for the naive approach (Algorithm 4) consists of three main steps. Step 1 generates all window-time interval pairs, $\langle S_i, T_i \rangle$ such that each spatial window is of a minimum size S_{min} and each time interval is of a minimum size t_{min}. The aggregated time series for each window-time interval pair is also calculated at this point. In Step 2, all window-time interval pairs whose average change rate exceeds the given threshold r are saved as potential candidates. Finally, Step 3 returns all candidates that are not subsets of any other candidate.

Algorithm 4 Naive PCW Algorithm

Input:

- A spatial field S with $M \times N$ locations that has a time series of values of length $|T|$,
- A threshold r of average change rate (ACR),
- A minimum region size S_{min},
- A minimum time length t_{min}

Output:

All persistent change windows $\langle S_i, T_i \rangle$ such that $|S_i| \geq S_{min}$ and $|T_i| \geq T_{min}$ and $\langle S_i, T_i \rangle$ is not a subset of any other pair $\langle S_j, T_j \rangle$ such that $S_i \subset S_j \wedge T_i \subset T_j$

Algorithm:

1: Enumerate all $\langle S_i, T_i \rangle$ (window, time interval) pairs and generate the aggregated time series for each $\langle S_i, T_i \rangle \in S \times T$ such that $S_i \subset S_j \wedge T_i \subset T_j$
2: $Candidates \leftarrow$ all $\langle S_i, T_i \rangle$ pairs whose ACR $\geq r$
3: **return** all $\langle S_i, T_i \rangle \in Candidates$ that are not subsets of any other $\langle S_j, T_j \rangle \in Candidates$

The main limitation of the naive approach is its high time complexity, where steps 1 and 2 require $O(M^3N^3T^3)$ and step 3 requires up to $O(M^4N^4T^4)$. The reason for this is that all ST windows are enumerated, and each pair needs to be compared to eliminate dominated ones, which gets expensive quickly. We expound on the details of time complexity and other theoretical properties in the Theoretical analysis section. Next, we describe our proposed approach.

4.3.2 The ST Window Enumeration and Pruning Approach

In order to reduce computational cost, we propose a space-time window enumeration and pruning (SWEP) approach, which enumerates all the ST windows in such an order

that no redundant evaluation is performed.

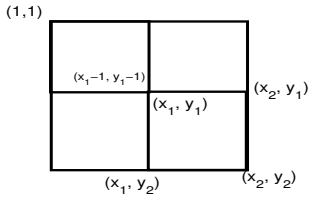


Figure 4.4: An illustration of the lookup table for spatial aggregate function

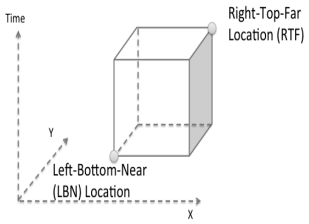


Figure 4.5: Left-Bottom-Near (LBN) and Right-Top-Far (RTF) representation of an ST window

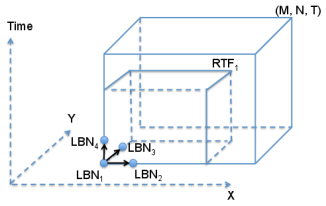
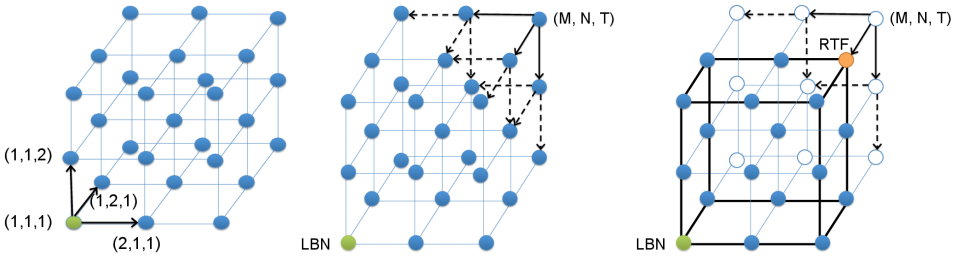


Figure 4.6: Illustration of the enumeration space for LBN

Fast computation of spatial aggregation: A primary optimization can be done to reduce the time for computing the interest measure. The spatial aggregation in each spatial window requires repeated scan of each location's value. We propose to precompute a lookup table so as to enable a constant-time lookup of the aggregate value for each spatial window and the ACR. The lookup table is three dimensional, with $N \times M \times T$ cells, where each cell (x, y, t) stores the SUM of spatial window $[(1, 1), (x, y)]$ at time t. The SUM of any spatial window $[(x_1, y_1), (x_2, y_2)]$ thus can be computed using four cells in the table in constant time: $\sum[(x_1, y_1), (x_2, y_2)] = \sum[(1, 1), (x_2, y_2)] - \sum[(1, 1), (x_1 - 1, y_2)] - \sum[(1, 1), (x_2, y_1 - 1)] + \sum[(1, 1), (x_1 - 1, y_1 - 1)]$. Figure 4.4 illustrates this relationship where the desired area is on the bottom-right. The lookup table also can be constructed based on this idea with a linear scan of each value, resulting in a $O(MNT)$ time cost. A similar idea has been used in image processing and computer vision (i.e., integral image [146]). In the following part, we focus on the enumeration and pruning of ST windows, and ignore the time cost of computing the ACR score in each pattern evaluation.

An alternative representation of ST window: An ST window $\langle S_i, T_i \rangle$ can be uniquely identified by the 3-dimensional coordinate of two locations, namely, the left-bottom-near (LBN) location, and the right-top-far (RTF) location. For example, an ST window $\langle [(1, 1), (10, 10)], (3, 4) \rangle$ can be represented as $\langle (1, 10, 3), (1, 10, 4) \rangle$. Figure 4.5 illustrates the LBN and RTF locations of an ST window.



(a) LBN enumeration of the SWEP algorithm (b) RTF enumeration with fixed LBN (c) Illustration of covered space for LBN

Figure 4.7: An illustration of Breadth-first enumeraten of RTF location (best viewed in color)

For simplicity when comparing two 3-D locations, we define the following relationship.

Definition 1. *Given two ST locations* $A = (x_1, y_1, t_1)$ *and* $B = (x_2, y_2, t_2)$, $A >> B$ $(B << A)$ *if* $x_1 \geq x_2 \wedge y_1 \geq y_2 \wedge t_1 \geq t_2$. *An ST windows* $W_1 = \langle LBN_1, RTF_1 \rangle$ *dominates all the ST windows* $W_2 = \langle LBN_2, RTF_2 \rangle$, *s.t.* $LBN_1 << LBN_2 \wedge RTF_2 >> RTF_1$

General idea of the SWEP algorithm: The SWEP algorithm enumerates all the ST windows by examining all pairs of LBN and RTF locations using a two-level enumeration process. The algorithm first enumerates all the LBN locations(the outer loop). For each LBN location, the algorithm enumerates all the valid RTF locations to find Persistent Change Window (PCW) (inner loop). The enumeration order is designed in such a way that (1) an ST window W is evaluated only when all of the ST windows W' are evaluated, where $W \subset W'$, and (2) if an ST window W is identified as a PCW pattern, then no subset of W should be evaluated.

In the outer loop, a breadth-first traversal is performed to enumerate the LBN location (x_1, y_1, t_1) of the candidate ST windows. For each LBN location, in the inner loop, the RTF location is also enumerated among a proper subset of all the locations. Each pair of $\langle LBN, RTF \rangle$ locations that form a PCW will be sent to output. The details of the algorithm are described below.

Enumerating LBN locations: The algorithm enumerates the LBN locations in a breadth-first manner, starting from the nearest LBN location (1,1,1). The enumeration space is all the ST locations in the dataset. This space can be modeled as a 3-D

directed lattice graph, where each node represents an ST location, and each directed edge connects neighboring ST locations with a one-unit difference in one of the three dimensions. The direction of each edge, aligned with one of the three dimensions, is from a location with a smaller coordinate to a location with a larger coordinate in that dimension. In such a lattice graph, each non-boundary node has three child nodes, one along each dimension, and three parent nodes, one along each dimension as well. A breadth-first enumeration guarantees that locations with smaller coordinates are always visited before locations with larger coordinates. Figure 4.7(a) illustrates the process of breadth-first enumeration of the LBN location.

Enumerating ST windows with a fixed LBN: We now consider the problem of enumerating ST windows with a fixed LBN location. In order to determine a unique ST window, we need both LBN and RTF locations. So we propose to enumerate all the valid RTF locations and evaluate the ST windows formed by the given LBN and each RTF. Similar to the LBN enumeration process, the RTF enumeration also can be viewed as a breadth-first traversal on the 3-D lattice graph. The traversal starts from the farthest RTF location (M, N, T), and proceeds towards the given LBN location. Figure 4.7(b) illustrates the traversal space and the process. The only difference of this traversal space from the previous LBN traversal space is that all the directed edges are in the opposite direction, i.e., pointing from a node to all its ST neighboring locations with smaller coordinates.

The enumeration rules are as follows: A node (candidate RTF) is enumerated only if all of its parent nodes are visited, and none of them form a PCW pattern with the given LBN node. The total number of parents of each node in the lattice graph will be 0 (root node), 1 (boundary nodes along two dimensions), 2 (boundary nodes along one dimension), or 3 (all inner nodes). If the candidate RTF node forms a PCW window with the current LBN node, this PCW will be output and none of the RTF node's children will be enumerated. For example, as illustrated in Figure 4.7(c), the orange node (RTF) and the LBN node form a PCW pattern. None of the blue nodes (successors of RTF) will be visited. Only non-successors of RTF (white nodes) are enumerated.

In addition, there are three conditions under which a node will not be enumerated (thus pruned). (1) the node to the left, lower, or near side of the given LBN node. This makes sure that all the ST windows have positive volume; (2) the corresponding

ST window formed by this node and the given LBN does not satisfy the minimum area or minimum time length constraints; (3) there exists another pair of LBN and RTF locations $\langle LBN_i, RTF_i\rangle$ such that $\langle LBN, RTF\rangle \subset \langle LBN_i, RTF_i\rangle$. The first two conditions are easy to check, while the third one is challenging. We show in the following part that the third condition can be guaranteed by pre-decide the enumeration space for each LBN.

The enumeration space of RTF for each LBN: To address the third challenge above, we consider the following scenario. As illustrated in Figure 4.6, when $LBN_1 = (x_1, y_1, t_1)$ is chosen as the LBN node (green node), we find a PCW pattern $W = \langle LBN_1, RTF_1\rangle$. In the next iteration, one of LBN_1's child node (e.g., LBN_2) is selected as the new LBN location. It is obvious that the RTF enumeration process now should not consider locations inside W, as the ST window they form will be certainly dominated by W.

We hereby define "enumeration space" for each LBN location as the subset of candidate RTF locations that should be examined in order to guarantee (3) above. The enumeration space of RTF for LBN_2 is $S \times T - W$ in this example. In a more general case, there might be more than one PCW generated with LBN_1 as the LBN location. The enumeration space of RTF for LBN_2 thus should be locations that are not inside any of these PCWs. Formally, the enumeration space of RTF for a particular LBN location can be represented as follows:

Lemma 1. *The enumeration space of RTF for a particular LBN location can be represented as the complement set of the union of all the PCWs, whose LBN location* $<<$ *the current LBN (a predecessor in LBN enumeration lattice graph). It also can be represented as the intersection of the complement sets to these PCWs. Formally, it can be express as follows:* $enumeration\_space(LBN) = S \times T - \bigcup\{PCW_i | PCW_i = \langle LBN_i, RTF_i\rangle, LBN_i << LBN\} = \bigcap\{S \times T - PCW_i | PCW_i = \langle LBN_i, RTF_i\rangle, LBN_i << LBN\}$

Proof. Suppose there exist a RTF location RTF_j in the enumeration space for the current LNB. If there exists some $PCW_i = \langle LBN_i, RTF_i\rangle$ such that $RTF_j \in PCW_i$, we have $RTF_i >> RTF_j$. Since PCW_i is already generated, LBN_i is visited before LBN in the enumeration. So $LBN_i << LBN$. So $\langle LBN, RTF_j\rangle \subset PCW_i$. This means RTF_j should not be part of any existing PCWs. The conclusion is proved. □

A careful review of the term $\{S \times T - PCW_i | PCW_i = \langle LBN_i, RTF_i \rangle, LBN_i << LBN\}$ may reveal that it is actually the set of locations examined but not selected to output in the RTF enumeration process for LBN_i. We define this set of location as the "covered space" of LBN_i. The covered space of each LBN can be obtained during the enumeration of RTF. For example, in the scenario illustrated in Figure 4.6, the node labeled RTF forms a PCW with LBN. The covered space of the current LBN (in green) is thus the nine blank nodes to the back of the grid structure. Children of LBN will only need to enumerate RTF among these locations to find PCW.

Due to the transitive property of the << relationship, and the nature of breadth-first enumeration of LBN locations, if $LBN_1 << LBN_2$, then $LBN_1 << LBN_2$'s children. This helps us simplify the representation of the enumeration space. Thus we finally derive the following lemma.

Lemma 2. *The enumeration space of RTF for a LBN location is the intersection of the covered space of all LBN's parents. Formally it can be written as:* $enumeration\_space(LBN) = \bigcap\{covered\_space(LBN_i) | LBN_i \in parents(LBN)\}$.

Proof. If $RTF_i \in enumeration\_space$(LBN), $RTF_i \in \bigcap covered\_space(LBN_j)$ where $LBN_j << LBN$ (Lemma 1). So $RTF_i \in covered\_space(every\ LBN_j << LBN)$. So $RTF_i \in covered\_space(LBN_i | LBN_i \in parent(LBN))$ This proves $LHS \subset RHS$. On the other hand, if $RTF_i \in RHS$, $RTF_i \notin$ any existing PCW (by definition). Thus, $RTF_i \in$ LHS. $RHS \subset RHS$, $LHS = RHS$. □

Algorithm 5 shows the process of the RTF enumeration with a fixed LBN location. The candidate enumeration space of RTF for each LBN location is implemented as a $M \times N \times T$ 0-1 array where 1 represents the location that needs to be enumerated. The algorithm also uses an array (nVst) to record the number of parents visited of each node. As described previously, if this number reaches the total number of parents (computed by $n\_parents()$), this node will be enumerated (Lines 14-18).

The pseudo code of the entire SWEP algorithm is presented in Algorithm 6. A list of 3-D arrays (C_Space) is kept as the covered space of each LBN that has been enumerated so far, where each array is a 0-1 array of size $M \times N \times T$. Since the covered space may be the same for different LBNs, another pointer array (Link_spc) is established to link each LBN to the corresponding version of the covered space. In

the outer loop, all the LBN locations are enumerated (Lines 4-16). In the inner loop, the RTF locations are enumerated (Line 8) by running Algorithm 5. Before entering the inner loop, the enumeration space (*e_space*) of LBN is generated by taking the intersection of the covered space of its parents (*Find_E_space*()). After the inner loop is finished, the *C_Space* list and *Link_spc* table are updated with the LBN's covered space (recorded while enumerating RTF). If it is same as an existing versions of covered space, just update the corresponding pointer in *Link_spc*. Otherwise add the new covered space to *C_space*.

4.4 Theoretical Analysis

In this section we analyze the correctness, completeness, and computational complexity of the propose algorithms.

Lemma 3. *The SWEP algorithm is correct. Correctness means that all the ST windows discovered by the algorithm are dominant PCWs based on the definitions.*

Proof. According Algorithm 6, each ST window in the output is evaluated against the threshold. Now we show that none of the ST windows in the output are subset of others. According to Lemma1 and Lemma2, this is guaranteed as long as the steps are followed. So the SWEP algorithm is correct. □

Lemma 4. *The SWEP algorithm is complete. The completeness means that all the PCWs which are not subsets of others in the given dataset are reported by the SWEP algorithm.*

Proof. This is to say that the SWEP algorithm only skip ST windows that are subset of other PCWs. This is obviously true for the enumeration of RTFs with a fixed LBN due to the nature of breadth first traversal. In the enumeration of different LBNs, according to the proof of Lemma 2, later generated PCWs will not be subset of any existing PCWs, as has been proved by the lemma. Obviously they can not be superset of earlier generated PCWs, either due to the order of their LBN nodes. This means the SWEP algorithm is complete. □

Time and Space complexity: The time complexity of the naive algorithm is $O(M^2N^2T^2)$ in the first two step. If considering the time for spatial aggregation, the

total complexity would reach $O(M^3N^3T^2)$. Suppose there are k PCWs generated, in the third step, the total time cost for eliminating the dominated PCWs is $O(k^2)$ (for pairwise comparison). In the worst case $k = O(M^2N^2T^2)$ and this complexity is $O(M^4N^4T^4)$. For space complexity, in the best case, no PCWs are generated. The space cost will be $O(1)$. In the worst case, all the k candidate PCWs need to be stored and the space complexity would reach $O(M^2N^2T^2)$.

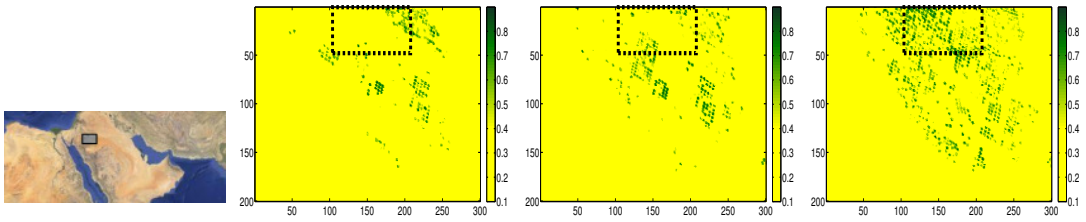
For the SWEP algorithm, the time complexity is O(1) in the best case, if the entire ST window is a PCW pattern. The algorithm terminates after the first evaluation. In the worst case, no PCW pattern exists and all of the ST windows need to be examined. This lead to $O(MNT)$ time to enumerate all the LBN locations, and $O(MNT)$ time in each round to enumerate all the RTF locations. The total complexity is $O(M^2N^2T^2)$. If we consider the time for computing the lookup table, the best case time complexity will be O(MNT). For space complexity, in the best case, all of the LBNs have the same covered space, which reduced the total memory cost to $O(MNT)$. In the worst case, each LBN will have a different covered_space, which lead to $O(M^2N^2T^2)$ memory cost.

Table 4.2: Comparison of time and space complexity of the two algorithms

	Naive approach	SWEP approach
Time complexity (best case)	$O(M^3N^3T^2)$	O(MNT)
Time complexity (worst case)	$O(M^4N^4T^4)$	$O(M^2N^2T^2)$
Space complexity (best case)	O(1)	$O(MNT)$
Space complexity (worst case)	$O(M^2N^2T^2)$	$O(M^2N^2T^2)$

4.5 Case Study

This section presents a case study of the proposed approach on a vegetation cover dataset. The goal of this section is to show that the SWEP approach can discover



(a) The study area in the map (b) Snapshot in 2001 (c) Snapshot in 2006 (d) Snapshot in 2012

Figure 4.8: The study area and one discovered PCW highlighted in three snapshots of the MODIS NDVI data

(a) Snapshot in 2001 (b) Snapshot in 2006 (c) Snapshot in 2012

Figure 4.9: Observations in the same area from Google Time lapse [1]

meaningful ST persistent change windows which corresponds to known phenomenon in particular areas.

4.5.1 Dataset and Settings:

In the case study, we use Normalized Difference of Vegetation Index (NDVI) data, measuring the vegetation cover extend, from the NASA MODIS project (MOD13Q1). The dataset has a spatial resolution of 250m, with a 16-day temporal resolution ranging from 2000 to 2012. The value ranges from 0 to 1 indicating more vegetation cover. We run the proposed algorithm on selected areas in Saudi Arabia to discover potential ST windows with a fast change of total vegetation cover. In order to get rid of the seasonality affect, we picked the snapshots of the same time of each year (July 27/28) and generated a spatial time series dataset. Figure 4.8 (a) shows the study area and its position in a world map. The data for this area is 200 by 300 pixels by 13 years.

In this example, we employ average as the spatial aggregate function in order to make the measure of different spatial windows comparable. It can be computed in the same way as we illustrated in Section 4.3. We select the average change rate (ACR)

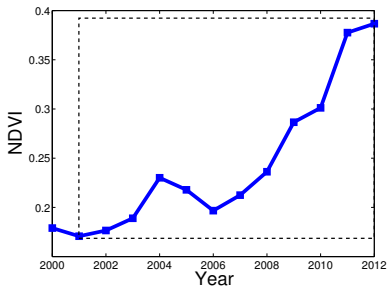


Figure 4.10: Spatial aggregated time series of the discovered PCW.

threshold as 0.1, meaning that the selected ST window should have a persistent increase of mean vegetation cover at an average rate of 10% per year. In order to reduce the trivial patterns discovered, we require that the PCWs should at least have 200 pixels in area and 4 years in time length.

4.5.2 Results: Irrigation in Saudi Arabia

The irrigation in Saudi Arabia has led to a significant increase in farmland and vegetation in recent decades. This process has been featured as one of the best-known land cover change process by Google Time-lapse [1]. Figure 4.9 (b-d) shows the snapshots of the observations from Google Time-lapse in 2001, 2006, and 2012 respectively.

The algorithm discovered a Persistent Change Window (PCW) consisting of (1) a rectangular spatial area with 100 by 50 pixels to the north of the study area, and (2) a time period from 2001 to 2012. using given threshold settings. The spatial footprint of the PCW is shown in Figure 4.8. The spatial aggregated (average) NDVI time series of this PCW is shown in Figure 4.10, in the highlighted box. The average change rate (ACR) of this pattern is 11.5% per year during the above period. This discovery shows that in the highlighted area, during 2001-2012, there was a persistent and rapid increase of vegetation cover. This result can also be verified by observations from Google Time-lapse shown in the previous figures, where a clear expansion of green land can be seen from 2001 through 2012. Note that there are also increase of NDVI out side the discovered PCW. However, due to our constraint on the minimum change rate, a larger PCW may not have a significant enough change rate to be displayed. But one could simply reset the threshold and discover these candidates.

The above results shows the effectiveness of the proposed approach in finding significant persistent change windows.

4.6 Experimental Evaluation

In this section, we present experimental evaluation results on a synthetic dataset. The goals of the experiments are (1) compare the time cost of the two algorithms with respect to dataset size, and (2) to compare their run time under different pattern distribution in the data.

4.6.1 Experiment Setup

Figure 4.11 illustrates the setup of the experiments. We implement the Naive algorithm and SWEP algorithm in a simulator. We use a synthetic dataset to feed the simulator.We manipulate the value distribution so that the size of the PCW can be controlled in each run. In the experiments, we vary the spatial window side length (M or N) and the time length (T) to test the scalability of the algorithms. In addition to the above parameters, we also use a new measure named Pattern Volume Ratio (PVR) to evaluate the performance under different data distribution. The Pattern Volume Ratio represents the ratio of (1) the volume of the largest WCP in the dataset, and (2) the volume of the entire dataset ($M \times N \times T$). For example, the PVR in the sample dataset given in Figure 4.2 is $(3 \times 3 \times 4)/(4 \times 4 \times 4) = 0.56$. This ratio is used as an indicator of the computational savings of SWEP algorithm, where PVR $\rightarrow 0$ indicates the worst case (PCWs are small) and PVR=1 indicates the best case (entire dataset is a PCW). All results are measured in seconds of CPU time.

The algorithms are implemented and tested in Matlab v2013a. The platform is a HP ProLiant BL280c G6 blade servers, with a quad-core 2.8 GHz Intel Xeon X5560 processor and 24 GB shared memory, running Linux system.

4.6.2 Results and Analysis

We first evaluate the computational time with respect to the size of the spatial window. We assume that the spatial window is a square (M=N), and vary the side length from 10 to 50. The area thus varies from 100 to 2500. The time length is fixed at 20.

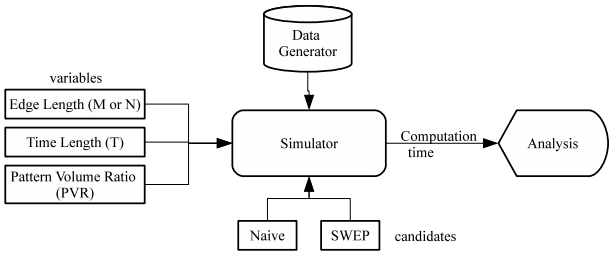
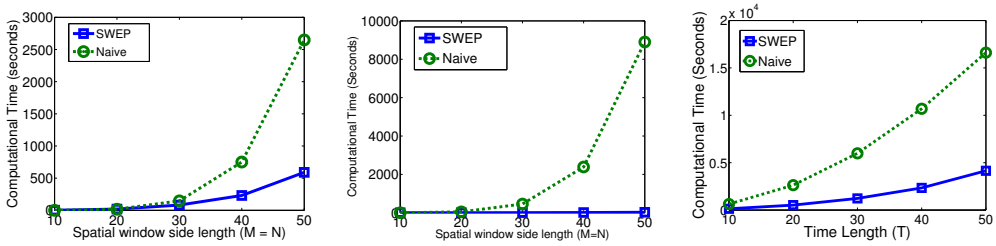


Figure 4.11: Experimental Setup

We first test a scenario close to the worst case, where the PVR value is fixed at 0.1. Figure 4.12(a) shows the computation time of the two algorithms. A clear trend can be observed. The run time of the naive algorithm increase at a much higher speed than the SWEP algorithm. The savings become significant after N$\geq$ 30 and reaches as much as 80% when N=50. We then test a scenario close to the best case, where the PVR value is fixed at 0.95. The largest PCW is almost as large as the entire dataset. Figure 4.12(b) shows the run time of the two algorithms. As can be seen, the run time of the naive algorithm increase exponentially, while the SWEP algorithm stays linear and orders of magnitude faster than the naive algorithm.

We next test the computational time of the two algorithms on increasing time length (T). We fix the spatial window as $50 \times 50 = 2500$, and increase the time length from 10 to 50. We also test the best case (PVR = 0.95) and the worst case (PVR = 0.1). As shown in Figure 4.12(c), in the worst case, the two algorithm both have subquadratic trends. The SWEP algorithm is always significantly faster than the naive algorithm, with up to 70% speedup. In the best case shown in Figure 4.12(d), however, the near-constant time SWEP is orders of magnitude faster than the the naive algorithm who increases linearly.

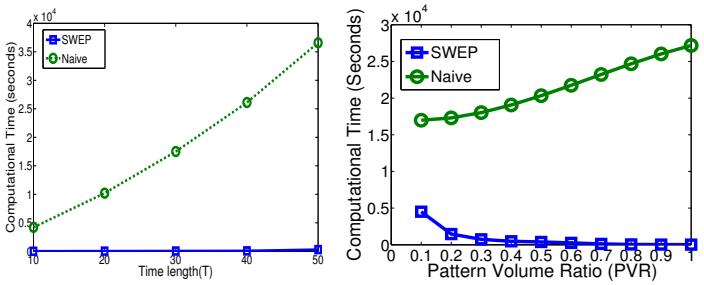
Finally, we evaluate the run time of the two algorithms on a fixed dataset with different PVR. Intuitively, a larger PVR favors the SWEP algorithm. The dataset is $50 \times 50 \times 50 = 125000$ locations. The PVR varies from 0.1 (near-worst case) to 1 (best case), with a step of 0.1. As can be seen in Figure 4.12(e), The run time of the SWEP algorithm keeps decreasing since the computational savings is increasing. However, the naive algorithm has a slightly increasing time since the total number of candidates increases, which lead to a longer time for Step 3 in Algorithm 4. The SWEP algorithm always outperforms the naive algorithm with huge computational savings.



(a) Computational time with varying window side length (M), PVR = 0.1

(b) Computational time with varying window side length (M), PVR = 0.95

(c) Computational time with varying time length (T), PVR = 0.1



(d) Computational time with varying time length (T), PVR = 0.95

(e) Computational time with varying PVR, M=N=T=50

Figure 4.12: Computational time comparison between the SWEP approach and the naive algorithm

4.7 Summary and Future Work

This work explored the problem of persistent change window (PCW) discovery. This problem is important for critical societal applications such as detecting desertification, deforestation, and urban sprawl. However, this problem is computationally challenging because of the large number of candidate patterns, the lack of monotonicity where sub-regions of a region of interest may not be interesting, the lack of predefined window sizes for region-time interval pairs, and large datasets of detailed resolution and high volume. We proposed an ST window enumeration and pruning approach (SWEP) as a computational solution to PCW. SWEP is novel because unlike previous approaches that focus on local spatial footprints, it uses zonal spatial footprints when finding region-time interval pairs such as deforestation in the Amazon over decades. Experiments on synthetic datasets showed that SWEP leads to computational savings over the naive approach without affecting result quality. We also presented a case study using vegetation

cover data to evaluate the effectiveness of SWEP and theoretical analysis to validate its correctness, completeness, and space/time complexity.

In future work, we would like to enhance our case studies by discovering more well recorded patterns (e.g., those shown by Google Time Lapse [1]) to validate the effectiveness of our approach. In addition, we would like to explore other algorithmic designs and cloud computing solutions to improve the efficiency of the current approach. We also plan to extend the current approach to discover interesting spatiotemporal patterns with irregular spatial footprints.

Algorithm 5 ST window Enumeration with fixed LBN

Input:

•A spatial field S with $M \times N$ locations that has a time series of values of length $|T|$,

• A threshold r of average change rate (ACR),

• A minimum window size S_{min}, A minimum time length t_{min},

• A LBN location (x_1, y_1, t_1)

Output:

All $W_i = \langle LBN, RTF_i \rangle$ pairs where $ACR(W_i) \geq r$ s.t. $|W_i|_s \geq S_{min}$ and $|W_i|_T \geq T_{min}$ and W_i is not a subset of any other ST window W_j such that $W_i \subset W_j$

Algorithm:

1: $Q_{RTF} \leftarrow (M, N, T)$, $nVst \leftarrow \{\mathbf{0}\}$
2: **while** Q_{RTF} is not empty **do**
3: $RTF_i = (x_i, y_i, t_i) \leftarrow Q_{RTF}.head()$
4: $Q_{RTF}.dequeue()$
5: **if** $(x_i - x_1) \cdot (y_i - y_1) \leq S_{min} || (t_i - t_1) \leq t_{min}$ **then**
6: Next loop;
7: **end if**
8: $Score \leftarrow$Compute_score($\langle LBN, RTF_i \rangle$)
9: **if** $Score \geq r$ **then**
10: Output $\langle LBN, RTF_i \rangle$
11: Next loop
12: **else**
13: **for** each child RTF_j of RTF_i **do**
14: **if** nVst[RTF_j]+1==n_parents(RTF_j) **then**
15: Q_{RTF}.enqueue(RTF_j)
16: **else**
17: nVst[RTF_j]++
18: **end if**
19: **end for**
20: **end if**
21: **end while**

Algorithm 6 ST Window Enumeration and Pruning (SWEP) Algorithm

Input:

- A spatial field S with $M \times N$ locations that has a time series of values of length $|T|$,
- A threshold r of average change rate (ACR),
- A minimum region size S_{min},
- A minimum time length t_{min}

Output:

All region-interval pairs $\langle S_i, T_i \rangle$ where $ACR(S_i, T_i) \geq r$ such that $|S_i| \geq S_{min}$ and $|T_i| \geq T_{min}$ and $\langle S_i, T_i \rangle$ is not a subset of any other pair $\langle S_j, T_j \rangle$ such that $S_i \subset S_j \wedge T_i \subset T_j$

Algorithm:

Scan all the windows with left-top corner (1,1) and build a lookup table
$Q_{LBN} \leftarrow (1, 1, 1)$, $C\_space[1] \leftarrow ones(M, N, T)$
$Link\_spc \leftarrow zeros(M, N, T)$
while Q_{LBN} is not empty **do**
    $LBN_i = (x_i, y_i, t_i) \leftarrow Q_{LBN}.head()$
    $Q_{LBN}.dequeue()$
    $e\_space_i \leftarrow$**Find_Enumerate_space**(LBN_i);
    Enumerate all the RTF in e_space_i
    Update C_space and $Link\_spc$
    Q_{LBN}.enqueue(LBN's unvisited children)
end while

Chapter 5

Conclusions and Future Work

This thesis explored the problem of spatiotemporal (ST) change footprint pattern discovery, a new pattern family in spatiotemporal big data analytics. Given a definition of change and a dataset about a spatiotemporal (ST) phenomenon, ST change footprint pattern discovery is the process of identifying the location and/or time of such changes in the data. ST change footprint pattern discovery is fundamentally important to a number of applications such as system monitoring, remote sensing image analysis, public health, public safety, environmental and climate science, etc. For example, finding ST change footprint in climate observations may help identify potential climate change events such as desertification, deforestation, and rainfall decrease.

However, ST change footprint pattern discovery is challenging due to the unknown scale of change patterns, lack of monotonicity of changes, and the large volume, variety and high cardinality of patterns in STBD.

In this thesis, we formally defined the ST change footprint pattern discovery problem as a new STBD analytics pattern discovery family. A novel taxonomy of change footprint patterns and related techniques was proposed for research gap analysis. Based on the analysis results, we further explored the computation solutions to the discovery of two raster-based change footprint patterns, namely, the interesting sub-path and persistent change windows (PCW). As for solution, we proposed an RTCP approach for the interesting sub-path discovery problem, and a space-time window enumeration and pruning (SWEP) approach for the PCW problem. A summary of the key results in this research are presented in Section 5.1 and future directions of this thesis are presented

in Section 5.2.

5.1 Key Results

This section presents a summary of the major results that were produced as a part of this thesis.

- **A taxonomy of change footprint pattern discovery techniques :** In survey of the change footprint pattern discovery techniques, we proposes a taxonomy of spatiotemporal change footprints that may be of use to researchers across multiple research domains. We built the taxonomy after conducting a multi-disciplinary review of research in ST change pattern discovery. Our taxonomy achieves two valuable goals. First, it classifies a wide variety of ST change footprints that have already received attention in different domains. Previously, much of this research was hidden from view, so to speak, due to the lack of common terminology across disciplines for discussing similar phenomena. Second, our taxonomy reveals gaps in the research, that is, change footprint patterns that have yet to be studied despite their potential applicability to many real-world problems. We especially note the need for research on ST change footprints on vector data.

- **Interesting interval/sub-path discovery:** We investigated the problem of discovering interesting sub-paths from a spatiotemporal path. In our recent work, we proposed a novel Sub-path Enumeration and Pruning (SEP) approach, with two computational designs for the problem. In this chapter, we model the problem as a Grid-based Directed Acyclic Graph (G-DAG) traversal problem, and present a new traversal algorithm, namely, the Row-wise traversal (after leaf-evaluation) with Column Pruning (RTCP) algorithm. Theoretical and experimental evaluation results show that the proposed RTCP algorithm is complete, correct, always runs much faster than the previous two design decisions with various pattern lengths, and achieves better scalability on both synthetic and real datasets. More importantly, RTCP has orders of magnitude less memory cost compared to the previous G-DAG traversal strategies in SEP.

- **ST Persistent Window Discovery:** We finally explored the problem of persistent change window (PCW) discovery. This problem is important for critical societal applications such as detecting desertification, deforestation, and urban sprawl. However, this problem is computationally challenging because of the large number of candidate patterns, the lack of monotonicity where sub-regions of a region of interest may not be interesting, the lack of predefined window sizes for region-time interval pairs, and large datasets of detailed resolution and high volume. We proposed an ST window enumeration and pruning approach (SWEP) as a computational solution to PCW. SWEP is novel because unlike previous approaches that focus on local spatial footprints, it uses zonal spatial footprints when finding region-time interval pairs such as deforestation in the Amazon over decades. Experiments on synthetic datasets showed that SWEP leads to computational savings over the naive approach without affecting result quality. We also presented a case study using vegetation cover data to evaluate the effectiveness of SWEP and theoretical analysis to validate its correctness, completeness, and space/time complexity.

5.2 Future Directions

My future research plan is to extend my current research and make efforts to develop new STBD analytic techniques as well as techniques on high-performance computing platforms to scale up the analytics for STBD.

5.2.1 STBD Analytics

Near-Term: In the near future, I would like to continue the exploration of interesting (change) sub-path and change window mining in the following aspects: (1) Investigate various interest measures to generalize the interesting sub-paths/intervals discovery framework for patterns in other applications, such as understanding the ST footprints of traffic congestion patterns using speed profiles. (2) Design efficient algorithms to accelerate persistent change window discovery. A multi-resolution enumeration and pruning paradigm could be designed to reduce the unnecessary evaluations of each individual candidate, as the interestingness measure may not change significantly if the

two candidate windows are very similar in the footprint.

Medium-term 1: Mining STBD for statistically significant change footprint patterns. Quality of patterns (e.g., interpretability) discovered from Spatiotemporal Big Data (STBD) is of great importance to the application domain. For example, vegetation in an area may exhibit natural variation over time due to the dynamics of earth systems. As a result, small changes may be observed in two or three consecutive years. However, such changes should not be flagged as a climate change event since they are very likely to be generated by random chance. The current paradigm of the ST change footprint pattern discovery is not designed to compute the statistical significance of patterns, leading to the risk of discovering non-interesting, randomly generated patterns. Computing statistically significant change footprint patterns is challenging due to (1) lack of suitable statistical measures of change for continuous variables in spatial field models, (2) large number of candidate patterns, and (3) expensive computation of statistical functions and tests (e.g., Monte-Carlo simulations).

The survey paper I wrote on ST change footprint pattern discovery technique [6] notes the key distinction between the above problem and the problems that the state-of-the-art statistical ST change detection techniques solve. The state-of-the-art techniques, such as space-time scan statistics [109, 107] and emerging cluster detection [114] are designed for point process datasets rather than raster datasets (e.g., climate data). I plan to investigate effective statistical measures and efficient pattern mining algorithms to address the gaps in related research.

Medium-term 2: Mining STDB for geographic features-based change footprint patterns. The current change footprint pattern discovery paradigm also assumes that patterns have simple shaped footprints, which is inadequate for many real scenarios. For example, the footprint of a forest fire may form a polygonal zone, and the growth of crops may appear as a linear pattern along a river. Finding geographic feature-based change footprint patterns is challenging due to the potentially huge enumeration space of candidates patterns. Most change footprint discovery techniques such as space-time scan statistics [109, 107]and emerging cluster detection [114] assume simple shapes (e.g., circles, rectangles) of patterns for the simplicity of computation. In our survey paper [6], we identified a few geographic features-based change patterns worth exploring. These footprints could be represented as spatial objects in the vector data model. Table 5.1

concludes the potential patterns for future exploration in bold with possible examples.

Table 5.1: Potential geographic feature-based ST change footprint patterns.

		Temporal footprint		
		Snapshot	Points in Time Series	Intervals in Time Series
Spatial footprint	Line segments	Spatial wobbling (significant zone boundaries)	Sales route grow over time	When did the river rapidly change its directions/routes?
	Polygon	Polygonal outliers/clusters	Abrupt change of jurisdiction shape	Gradual change of market footprints
	Spatial network	Intersections with abrupt traffic speed change	Year of significant growth of the traffic network	Periods of rapid growth of rail road network?

To address these limitations, I plan to investigate a more flexible representation of change footprints and design efficient pattern discovery algorithms.

5.2.2 STBD Platforms

I also plan to investigate highly scalable computation techniques on parallel and cloud platforms to speed up the discovery of change footprint patterns on STBD. The non-iterative nature of the change footprint discovery problem makes it possible to take advantage of parallel computing platforms to accelerate the algorithms. Our preliminary work [147] shows that by using CUDA we can achieve at least 20 times of speedup on a dataset with 600000 longitudinal paths. The main challenge in this work is to partition the data for load balancing without losing patterns across different partitions. Another challenge is removing dominated patterns (e.g., subset of a long dominant interesting interval) due to the limited shared memory of parallel computing platforms (e.g., CUDA).

For near-term, I plan to further investigate the parallel computational structure of the RTCP algorithm for the interesting sub-path/interval discovery problem. I would also like to investigate the parallel computational structure of the persistent window discovery (PCW) problem on platforms such as CUDA.

For medium-term, I plan to investigate effective load balancing and parallel computational structures for the statistically significant pattern discovery and the complex-shaped pattern discovery problems. I also plan to investigate solutions to the above STBD analytic problems on cloud computing platforms (e.g., Hadoop MapReduce). I also plan to develop high-performance STBD analytical platforms (e.g., GIS software/packages) to incorporate the above techniques.

5.2.3 Long-term Goals

For the long-term goals beyond five years, I plan to establish novel research in different aspects of STBD analytics, such as defining new pattern families, new analytical dimensions (e.g., multi-scales spatial relationship analysis), and exploring new applications (e.g., public health, public safety, transportation sciences, and social science) of the above innovative ideas. More importantly, I will pursue to develop physics-aware data analytics and knowledge discovery techniques. Current data driven techniques (e.g., data mining and knowledge discovery) are heavily focused on analyzing the characteristics of the data, without incorporating rules and common senses from the physical world in the computation. The disadvantage of this is that some results may not make sense in reality, and human assistance is required to eliminate them using domain knowledge. In my future plan, I will pursue to balance physical understanding and data analytics by developing automated and efficient computational techniques that incorporate physical rules and understandings into the analytics to improve patterns interpretability and to reduce human involvement. Progression in this work may potentially benefit a wide range of societal applications beyond climate and earth science in the future.

References

[1] Google Earth Engine (Accessed: June 22, 2013).

[2] Computing Community Consortium (CCC). From GPS and Virtual Globes to Spatial Computing 2020: The Next Transformative Technology. A Community Whitepaper resulting from the 2012 CCC Spatial Computing 2020 Workshop, Sep., 2012. Available at: `www.cra.org/ccc/files/docs/Spatial_Computing_Report-2013.pdf`.

[3] James Manyika, Michael Chui, Brad Brown, Jacques Bughin, Richard Dobbs, Charles Roxburgh, and Angela Hung Byers. Big data: The next frontier for innovation, competition and productivity. Technical report, McKinsey Global Institute, 2011.

[4] Executive Order: Preparing the United States for the Impacts of Climate Change. The White House, Nov. 01, 2013. Available at: `www.whitehouse.gov/the-press-office/2013/11/01/executive-order-preparing-united-states-impacts-climate-change`.

[5] Wikipedia. Foursquare — wikipedia, the free encyclopedia, 2014. [Online; accessed 14-May-2014].

[6] Xun Zhou, Shashi Shekhar, and Reem Y Ali. Spatiotemporal change footprint pattern discovery: an inter-disciplinary survey. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 4(1):1–23, 2014.

[7] Xun Zhou, Shashi Shekhar, Pradeep Mohan, Stefan Liess, and Peter K Snyder. Discovering interesting sub-paths in spatiotemporal datasets: A summary of results. In *Proceedings of the 19th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, pages 44–53. ACM, 2011.

[8] Xun Zhou, Shashi Shekhar, Pradeep Mohan, Stefan Liess, and Peter K Snyder. Discovering Interesting Sub-paths in Spatiotemporal Datasets. *IEEE Transection on Data and Knowledge Engineering (TKDE), under review.*

[9] Xun Zhou, Shashi Shekhar, and Dev Oliver. Discovering persistent change windows in spatiotemporal datasets: A summary of results. In *Proc. 2nd ACM SIGSPATIAL International Workshop on Analytics for Big Geospatial Data (BigSpatial-2013)*, Orlando, FL, Nov. 5, 2013. ACM.

[10] George EP Box, Alberto Luceno, and María del Carmen Paniagua-Quiñones. *Statistical control by monitoring and adjustment*, volume 898. John Wiley & Sons, 2011.

[11] Ross S Lunetta, Christopher D Elvidge, et al. *Remote sensing change detection: environmental monitoring methods and applications.* Taylor & Francis Ltd, 1999.

[12] A. Singh. Review article digital change detection techniques using remotely-sensed data. *International journal of remote sensing*, 10(6):989–1003, 1989.

[13] J.R. Eastman and J. McKendry. *Change and Time Series Analysis in GIS.* UNITAR, Geneva.

[14] M. Kulldorff and N. Nagarwalla. Spatial disease clusters: detection and inference. *Statistics in medicine*, 14(8):799–810, 1995.

[15] S. Chainey, L. Tompson, and S. Uhlig. The utility of hotspot mapping for predicting spatial patterns of crime. *Security Journal*, 21(1):4–28, 2008.

[16] W.F. Fagan, M.J. Fortin, and C. Soykan. Integrating edge detection and dynamic modeling in quantitative analyses of ecological boundaries. *BioScience*, 53(8):730–738, 2003.

[17] James Hansen, Makiko Sato, and Reto Ruedy. Perception of climate change. *Proceedings of the National Academy of Sciences*, 109(37):E2415–E2423, 2012.

[18] Michèle Basseville, Igor V Nikiforov, et al. *Detection of abrupt changes: theory and application*, volume 104. Prentice Hall Englewood Cliffs, 1993.

[19] SA Shaban. Change point problem and two-phase regression: an annotated bibliography. *International statistical review*, 48(1):83–93, 1980.

[20] S. Zacks. Classical and bayesian approaches to the change-point problem: Fixed sample and sequential procedures. Report, DTIC Document, 1982.

[21] Jie Chen and Arjun K Gupta. On change point detection and estimation. *Communications in statistics-simulation and computation*, 30(3):665–697, 2001.

[22] P.C., I. Jonckheere, K. Nackaerts, B. Muys, and E. Lambin. Digital change detection methods in ecosystem monitoring: a review. *International journal of remote sensing*, 25(9):1565–1596, 2004.

[23] M. Barkat. Signal detection and estimation. 2005.

[24] D Lu, P Mausel, E Brondizio, and E Moran. Change detection techniques. *International journal of remote sensing*, 25(12):2365–2401, 2004.

[25] W.K. Wong and D.B. Neill. Tutorial on event detection. *Presentation in ACM SIGKDD Conference on Knowledge Discovery and Data Mining 2009. http://www.pptsearch.net/download.php?fid=109746.*

[26] W.H. Womble. Differential systematics. *Science*, 114(2961):315, 1951.

[27] S. Banerjee and A.E. Gelfand. Bayesian wombling: Curvilinear gradient assessment under spatial process models. *Journal of the American Statistical Association*, 101:1487–1501, 2006.

[28] H. Lu and B.P. Carlin. Bayesian areal wombling for geographical boundary analysis. *Geographical Analysis*, 37(3):265–285, 2005.

[29] Neal L Oden, Robert R Sokal, Marie?Jose Fortin, and Hans Goebl. Categorical wombling: detecting regions of significant change in spatially located categorical variables. *Geographical Analysis*, 25(4):315–336, 1993.

[30] S. Shekhar, M.R. Evans, J.M. Kang, and P. Mohan. Identifying patterns in spatial information: A survey of methods. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 1(3):193–214, 2011.

[31] Joint Institute for the Study of the Atmosphere and Ocean(JISAO). Sahel rainfall index. `http://jisao.washington.edu/data/sahel/`.

[32] Tucker, C.J., J.E. Pinzon, M.E. Brown. Global inventory modeling and mapping studies. Global Land Cover Facility, University of Maryland, College Park, Maryland, 1981-2006.

[33] C.J. Tucker, J.E. Pinzón, M.E. Brown, D.A. Slayback, E.W. Pak, R. Mahoney, E.F. Vermote, and N. El Saleous. An extended AVHRR 8-km NDVI dataset compatible with MODIS and SPOT vegetation NDVI data. *International Journal of Remote Sensing*, 26(20):4485–4498, 2005.

[34] J Pinzon, Molly E Brown, and Compton J Tucker. Satellite time series correction of orbital drift artifacts using empirical mode decomposition. *Hilbert-Huang transform: introduction and applications*, (Part II):167–186, 2005.

[35] S. Shekhar and S. Chawla. *Spatial databases: a tour*, volume 1. Prentice Hall, 2003.

[36] M. Worboys and M. Duckham. *GIS: A computing perspective*. CRC, ISBN:0415283752., 2004.

[37] CIA World Factbook. `https://www.cia.gov/library/publications/the-world-factbook/`.

[38] S. Boriah, V. Kumar, M. Steinbach, C. Potter, and S. Klooster. Land cover change detection: a case study. In *Proceeding of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 857–865. ACM, 2008.

[39] Pusheng Zhang, Yan Huang, Shashi Shekhar, and Vipin Kumar. Correlation analysis of spatial time series datasets: A filter-and-refine approach. *Advances in Knowledge Discovery and Data Mining*, pages 563–563, 2003.

[40] Walter R Gilks, Sylvia Richardson, and David Spiegelhalter. *Markov Chain Monte Carlo in practice: interdisciplinary statistics*, volume 2. Chapman & Hall/CRC, 1995.

[41] Waldo R Tobler. A computer movie simulating urban growth in the detroit region. *Economic geography*, 46:234–240, 1970.

[42] S. Banerjee, B.P. Carlin, and A.E. Gelfand. *Hierarchical modeling and analysis for spatial data*, volume 101. Chapman & Hall, 2004.

[43] V. Chandola, D. Hui, L. Gu, B. Bhaduri, and R.R. Vatsavai. Using Time Series Segmentation for Deriving Vegetation Phenology Indices from MODIS NDVI Data. In *2010 IEEE International Conference on Data Mining Workshops*, pages 202–208. IEEE, 2010.

[44] Valery Guralnik and Jaideep Srivastava. Event detection from time series data. In *Proceedings of the fifth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 33–42. ACM, 1999.

[45] Yi Fang, Auroop R Ganguly, Nagendra Singh, Veeraraghavan Vijayaraj, Neal Feierabend, and David T Potere. Online change detection: Monitoring land cover from remotely sensed data. In *Data Mining Workshops, 2006. ICDM Workshops 2006. Sixth IEEE International Conference on*, pages 626–631. IEEE.

[46] Zhe Zhu, Curtis E Woodcock, and Pontus Olofsson. Continuous monitoring of forest disturbance using all available landsat imagery. *Remote Sensing of Environment*, 2012, 122:75–91.

[47] Edith Gabriel and Denis Allard. Evaluating the sampling pattern when detecting zones of abrupt change. *Environmental and Ecological Statistics*, 15(4):469–489, 2008.

[48] Robert R Sokal and Barbara A Thomson. Spatial genetic structure of human populations in japan. *Human biology*, 70(1):1, 1998.

[49] Jean-Pierre Bocquet-Appel and Lucienne Jakobi. Barriers to the spatial diffusion for the demographic transition in western europe. *Spatial analysis of biodemographic data*, 16:117–129, 1996.

[50] G. Barbujani, N.L. Oden, and R.R. Sokal. Detecting regions of abrupt change in maps of biological variables. *Systematic Biology*, 38(4):376–389, 1989.

[51] Marie-Josee Fortin. Edge detection algorithms for two-dimensional ecological data. *Ecology*, pages 956–965, 1994.

[52] David L Strayer, Mary E Power, William F Fagan, Steward TA Pickett, and Jayne Belnap. A classification of ecological boundaries. *BioScience*, 53(8):723–729, 2003.

[53] David H Douglas and Thomas K Peucker. Algorithms for the reduction of the number of points required to represent a digitized line or its caricature. *Cartographica: The International Journal for Geographic Information and Geovisualization*, 10(2):112–122, 1973.

[54] Y. Kawahara and M. Sugiyama. Change-point detection in time-series data by direct density-ratio estimation. In *Proceedings of 2009 SIAM International Conference on Data Mining (SDM2009)*, pages 389–400, 2009.

[55] Venkata K Jandhyala, Stergios B Fotopoulos, and Douglas M Hawkins. Detection and estimation of abrupt changes in the variability of a process. *Computational statistics & data analysis*, 40(1):1–19, 2002.

[56] Ryan Prescott Adams and David JC MacKay. Bayesian online changepoint detection. *arXiv preprint arXiv:0710.3742*, 2007.

[57] Douglas A Wolfe and Edna Schechtman. Nonparametric statistical procedures for the changepoint problem. *Journal of Statistical Planning and Inference*, 9(3):389–396, 1984.

[58] Herman Chernoff and Shelemyahu Zacks. Estimating the current mean of a normal distribution which is subjected to changes in time. *The Annals of Mathematical Statistics*, pages 999–1018, 1964.

[59] Douglas M Hawkins. Fitting multiple change-point models to data. *Computational Statistics & Data Analysis*, 37(3):323–341, 2001.

[60] D. Barry and J.A. Hartigan. A bayesian analysis for change point problems. *Journal of the American Statistical Association*, pages 309–319, 1993.

[61] David Siegmund. Boundary crossing probabilities and statistical applications. *The Annals of Statistics*, 14(2):361–404, 1986.

[62] PR Krishnaiah and BQ Miao. 19 review about estimation of change points. *Handbook of Statistics*, 7:375–402, 1988.

[63] M. Csorgo and Lajos Horvath. 20 nonparametric methods for change point problems. *Handbook of statistics*, 7:403–425, 1988.

[64] Tze Leung Lai. Sequential changepoint detection in quality control and dynamical systems. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 613–658, 1995.

[65] M Basseville. Detecting changes in signals and systems a survey. *Automatica*, 24(3):309–326, 1988.

[66] Gouri K Bhattacharyya. 5 tests of randomness against trend or serial correlations. *Handbook of statistics*, 4:89–111, 1984.

[67] Fredrik Gustafsson and Fredrik Gustafsson. *Adaptive filtering and change detection*, volume 1. Wiley Londres, 2000.

[68] Boris E Brodsky and Boris S Darkhovsky. *Nonparametric methods in change point problems*, volume 243. Springer, 1993.

[69] E. S. Page. Continuous inspection schemes. *Biometrika*, 41(1/2):100–115, 1954.

[70] Jan Kucera, Paulo Barbosa, and Peter Strobl. Cumulative sum charts-a novel technique for processing daily time series of modis data for burnt area mapping in portugal. In *Analysis of Multi-temporal Remote Sensing Images, 2007. MultiTemp 2007. International Workshop on the*, pages 1–6. IEEE, 2007.

[71] ZM Nopiah, MN Baharin, S Abdullah, MI Khairir, and CKE Nidzwan. The detection of abrupt changes in fatigue data by using cumulative sum (cusum) method. In *Recent Advances in Applied and Theoretical Mechanics: Proceedings of the 4th International Conference on Applied and Theoretical Mechanics (Mechanics' 08)*, pages 75–80.

[72] Ella Bingham, Aristides Gionis, Niina Haiminen, Heli Hiisil, Heikki Mannila, and Evimaria Terzi. Segmentation and dimensionality reduction. In *2006 SIAM Conference on Data Mining*, pages 372–383, 2006.

[73] Johan Himberg, Kalle Korpiaho, Heikki Mannila, Johanna Tikanmaki, and Hannu TT Toivonen. Time series segmentation for context recognition in mobile devices. In *Data Mining, 2001. ICDM 2001, Proceedings IEEE International Conference on*, pages 203–210. IEEE.

[74] Eamonn Keogh, Selina Chu, David Hart, and Michael Pazzani. An online algorithm for segmenting time series. In *Proceedings of IEEE International Conference on Data Mining (ICDM 2001), 2001*, pages 289–296. IEEE.

[75] Eamonn Keogh, Selina Chu, David Hart, and Michael Pazzani. Segmenting time series: A survey and novel approach. *Data mining in time series databases*, 57:1–22, 2004.

[76] Hagit Shatkay and Stanley B Zdonik. Approximate queries and representations for large data sequences. In *Data Engineering, 1996. Proceedings of the Twelfth International Conference on*, pages 536–545. IEEE.

[77] Chung-Sheng Li, Philip S Yu, and Vittorio Castelli. Malm: a framework for mining sequence database at multiple abstraction levels. In *Proceedings of the seventh international conference on Information and knowledge management*, pages 267–272. ACM.

[78] Eamonn J Keogh and Michael J Pazzani. An enhanced representation of time series which allows fast and accurate classification, clustering and relevance feedback. In *KDD*, volume 98, pages 239–243, 1998.

[79] G.T. Narisma, J.A. Foley, R. Licker, and N. Ramankutty. Abrupt changes in rainfall during the twentieth century. *Geophysical Research Letters*, 34(6):L06710, 2007.

[80] A. Dai, P.J. Lamb, K.E. Trenberth, M. Hulme, P.D. Jones, and P. Xie. The recent sahel drought is real. *International Journal of Climatology*, 24(11):1323–1331, 2004.

[81] Florentin Bujor, Emmanuel Trouv, Lionel Valet, J-M Nicolas, and J-P Rudant. Application of log-cumulants to the detection of spatiotemporal discontinuities in multitemporal sar images. *Geoscience and Remote Sensing, IEEE Transactions on*, 42(10):2073–2084, 2004.

[82] Yukio Kosugi, Mitsuteru Sakamoto, Munenori Fukunishi, Wei Lu, Takeshi Doihara, and Shigeru Kakumoto. Urban change detection related to earthquakes using an adaptive nonlinear mapping of high-resolution images. *Geoscience and Remote Sensing Letters, IEEE*, 1(3):152–156, 2004.

[83] Gerardo Di Martino, Antonio Iodice, Daniele Riccio, and Giuseppe Ruello. A novel approach for disaster monitoring: fractal models and tools. *Geoscience and Remote Sensing, IEEE Transactions on*, 45(6):1559–1570, 2007.

[84] J. Im and J.R. Jensen. A change detection model based on neighborhood correlation image analysis and decision tree classification. *Remote Sensing of Environment*, 99(3):326–340, 2005.

[85] Yakoub Bazi, Lorenzo Bruzzone, and Farid Melgani. An unsupervised approach based on the generalized gaussian model to automatic change detection in multitemporal sar images. *Geoscience and Remote Sensing, IEEE Transactions on*, 43(4):874–887, 2005.

[86] Jordi Inglada and Grgoire Mercier. A new statistical similarity measure for change detection in multitemporal sar images and its extension to multiscale change analysis. *Geoscience and Remote Sensing, IEEE Transactions on*, 45(5):1432–1445, 2007.

[87] Francesca Bovolo and Lorenzo Bruzzone. A split-based approach to unsupervised change detection in large-size multitemporal images: Application to tsunami-damage assessment. *Geoscience and Remote Sensing, IEEE Transactions on*, 45(6):1658–1670, 2007.

[88] Turgay Celik. Change detection in satellite images using a genetic algorithm approach. *Geoscience and Remote Sensing Letters, IEEE*, 7(2):386–390, 2010.

[89] Rouhollah Dianat and Shohreh Kasaei. Change detection in optical remote sensing images using difference-based methods and spatial information. *Geoscience and Remote Sensing Letters, IEEE*, 7(1):215–219, 2010.

[90] Gabriele Moser, Elena Angiati, and Sebastiano B Serpico. Multiscale unsupervised change detection on optical images by markov random fields and wavelets. *Geoscience and Remote Sensing Letters, IEEE*, 8(4):725–729, 2011.

[91] R.J. Radke, S. Andra, O. Al-Kofahi, and B. Roysam. Image change detection algorithms: a systematic survey. *Image Processing, IEEE Transactions on*, 14(3):294–307, 2005.

[92] R. Thoma and M. Bierling. Motion compensating interpolation considering covered and uncovered background. *Signal Processing: Image Communication*, 1(2):191–212, 1989.

[93] T. Aach and A. Kaup. Bayesian algorithms for adaptive change detection in image sequences using markov random fields. *Signal Processing: Image Communication*, 7(2):147–160, 1995.

[94] Gang Chen, Geoffrey J Hay, Luis MT Carvalho, and Michael A Wulder. Object-based change detection. *International Journal of Remote Sensing*, 33(14):4434–4457, 2012.

[95] Baudouin Desclee, Patrick Bogaert, and Pierre Defourny. Forest change detection by statistical object-based method. *Remote Sensing of Environment*, 102(1):1–11, 2006.

[96] J Im, JR Jensen, and JA Tullis. Object?based change detection using correlation image analysis and image segmentation. *International Journal of Remote Sensing*, 29(2):399–423, 2008.

[97] Lorenzo Bruzzone and D Fernandez Prieto. An adaptive semiparametric and context-based approach to unsupervised change detection in multitemporal remote-sensing images. *Image Processing, IEEE Transactions on*, 11(4):452–466, 2002.

[98] Remote sensing and geospatial analysis laboratory, university of minnesota, twin cities metro area impervious surface data 1998 and 2002.

[99] Til Aach, Andr Kaup, and Rudolf Mester. Statistical model-based change detection in moving video. *Signal processing*, 31(2):165–180, 1993.

[100] Eric JM Rignot and Jacob J van Zyl. Change detection techniques for ers-1 sar data. *Geoscience and Remote Sensing, IEEE Transactions on*, 31(4):896–906, 1993.

[101] Y. Yakimovsky. Boundary and object detection in real world images. *Journal of the ACM (JACM)*, 23(4):599–618, 1976.

[102] S. Liang, S. Banerjee, and B.P. Carlin. Bayesian wombling for spatial point processes. *Biometrics*, 65(4):1243–1253, 2009.

[103] M. Kulldorff. A spatial scan statistic. *Communications in statistics-theory and methods*, 26(6):1481–1496, 1997. Xc912 Times Cited:716 Cited References Count:22.

[104] M. Kulldorff, L. Huang, and K. Konty. A scan statistic for continuous data based on the normal probability model. *International journal of health geographics*, 8(1):58, 2009.

[105] L. Huang, M. Kulldorff, and D. Gregorio. A spatial scan statistic for survival data. *Biometrics*, 63(1):109–118, 2007.

[106] I. Jung, M. Kulldorff, and A.C. Klassen. A spatial scan statistic for ordinal data. *Statistics in Medicine*, 26(7):1594–1607, 2007.

[107] Martin Kulldorff, Richard Heffernan, Jessica Hartman, Renato Assunçao, and Farzad Mostashari. A space–time permutation scan statistic for disease outbreak detection. *PLoS medicine*, 2(3):e59, 2005.

[108] M. Kulldorff, WF Athas, EJ Feurer, BA Miller, and CR Key. Evaluating cluster alarms: a space-time scan statistic and brain cancer in los alamos, new mexico. *American Journal of Public Health*, 88(9):1377–1380, 1998.

[109] M. Kulldorff. Prospective time periodic geographical disease surveillance using a scan statistic. *Journal of the Royal Statistical Society: Series A (Statistics in Society)*, 164(1):61–72, 2001.

[110] Marcelo A Costa, Martin Kulldorff, and Renato M Assuncao. A space time permutation scan statistic with irregular shape for disease outbreak detection. *Advances in Disease Surveillance*, 4:243, 2007.

[111] D.B. Neill and A.W. Moore. Rapid detection of significant spatial clusters, 2004.

[112] D. Neill, A. Moore, and G. Cooper. A bayesian spatial scan statistic. *Advances in neural information processing systems*, 18:1003, 2006.

[113] M Kulldorff. Satscan user guide for version 7.0. *Accessed January*, 18:2008, 2006.

[114] D.B. Neill, A.W. Moore, M. Sabhnani, and K. Daniel. Detection of emerging space-time clusters, 2005.

[115] Amy Hillier and Anne Kelly Knowles. *Placing history: how maps, spatial data, and GIS are changing historical scholarship*. ESRI press, 2008.

[116] Merrick Lex Berman and China Historical GIS. A data model for historical gis: The chgis time series. *Cambridge, MA, Harvard Yenching Institute Technical Report*, 2003.

[117] Harvard university and fudan university. the china histocial gis project.

[118] University of nebraska-lincoln. railroads and the making of modern america, 2011.

[119] William G Thomas. *The Iron Way: Railroads, the Civil War, and the Making of Modern America*. Yale University Press, 2011.

[120] University of nebraska-lincoln. historical gis: The 1840-1845-1850-1861-1870 railroad system in america, state and national shapefiles, 2011.

[121] I.R. Noble. A model of the responses of ecotones to climate change. *Ecological Applications*, 3(3):396–403, 1993.

[122] D. Nikovski and A. Jain. Fast adaptive algorithms for abrupt change detection. *Machine learning*, 79(3):283–306, 2010.

[123] M. Sharifzadeh, F. Azmoodeh, and C. Shahabi. Change detection in time series data using wavelet footprints. *Advances in Spatial and Temporal Databases*, pages 127–144, 2005.

[124] Jun-ichi Takeuchi and Kenji Yamanishi. A unifying framework for detecting outliers and change points from time series. *Knowledge and Data Engineering, IEEE Transactions on*, 18(4):482–492, 2006.

[125] J. Canny. A computational approach to edge detection. *Readings in computer vision: issues, problems, principles, and paradigms*, 184(87-116):86, 1987.

[126] S. Shekhar and S. Chawla. Spatial databases: A tour. prentice hall, 2003 (isbn 013-017480-7).

[127] S. Cluet and G. Moerkotte. Efficient evaluation of aggregates on bulk types. In *In Proc. Int. Workshop on Database Programming Languages*, 1995.

[128] Thomas L Delworth, Anthony J Broccoli, Anthony Rosati, Ronald J Stouffer, V Balaji, John A Beesley, William F Cooke, Keith W Dixon, John Dunne, KA Dunne, et al. Gfdl's cm2 global coupled climate models. part i: Formulation and simulation characteristics. *Journal of Climate*, 19(5):643–674, 2006.

[129] Rolf H Reichle, Randal D Koster, Gabriëlle JM De Lannoy, Barton A Forman, Qing Liu, Sarith PP Mahanama, and Ally Tour. Assessment and enhancement of merra land surface hydrology estimates. *Journal of climate*, 24(24), 2011.

[130] J.E. Janowiak. An investigation of interannual rainfall variability in africa. *Journal of Climate*, 1:240–255, 1988.

[131] Alessandra Giannini, R Saravanan, and P Chang. Oceanic forcing of sahel rainfall on interannual to interdecadal time scales. *Science*, 302(5647):1027–1030, 2003.

[132] CK Folland, TN Palmer, and DE Parker. Sahel rainfall and worldwide sea temperatures, 1901–85. *Nature*, 320(6063):602–607, 1986.

[133] J.M. Kang, S. Shekhar, C. Wennen, and P. Novak. Discovering Flow Anomalies: A SWEET Approach. In *Data Mining, 2008. ICDM'08. Eighth IEEE International Conference on*, pages 851–856. IEEE, 2009.

[134] J.G. Lee, J. Han, and X. Li. Trajectory outlier detection: A partition-and-detect framework. In *Data Engineering, 2008. ICDE 2008. IEEE 24th International Conference on*, pages 140–149. IEEE.

[135] Lasse Bergroth, Harri Hakonen, and Timo Raita. A survey of longest common subsequence algorithms. In *String Processing and Information Retrieval, 2000. SPIRE 2000. Proceedings. Seventh International Symposium on*, pages 39–48. IEEE, 2000.

[136] Christos Faloutsos, M. Ranganathan, and Yannis Manolopoulos. Fast subsequence matching in time-series databases. In *SIGMOD Conference*, pages 419–429, 1994.

[137] Donald Berndt and James Clifford. Using dynamic time warping to find patterns in time series. In *KDD workshop*, volume 10, pages 359–370. Seattle, WA, 1994.

[138] Eamonn Keogh. Exact indexing of dynamic time warping. In *Proceedings of the 28th international conference on Very Large Data Bases*, pages 406–417. VLDB Endowment, 2002.

[139] Lei Chen, M Tamer Özsu, and Vincent Oria. Robust and fast similarity search for moving object trajectories. In *Proceedings of the 2005 ACM SIGMOD international conference on Management of data*, pages 491–502. ACM, 2005.

[140] Joachim Gudmundsson and Marc van Kreveld. Computing longest duration flocks in trajectory data. In *Proceedings of the 14th annual ACM international symposium on Advances in geographic information systems*, pages 35–42. ACM, 2006.

[141] World deforestation rates and forest cover statistics, 2000-2005, (Accessed: June 22, 2013).

[142] P.C. Coppin, I. Jonckheere, K. Nackaerts, B. Muys, and E. Lambin. Digital change detection methods in ecosystem monitoring: a review. *International journal of remote sensing*, 25(9):1565–1596, 2004.

[143] J-F Mas. Monitoring land-cover changes: a comparison of change detection techniques. *International journal of remote sensing*, 20(1):139–152, 1999.

[144] ERDAS Imaging (Accessed: August 30, 2013).

[145] ILWIS - Remote Sensing and GIS software (Accessed: August 30, 2013).

[146] Paul Viola and Michael Jones. Rapid object detection using a boosted cascade of simple features. In *Proc. 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2001)*, volume 1, pages I–511. IEEE, 2001.

[147] Sushil K Prasad, Shashi Shekhar, Michael McDermott, Xun Zhou, Michael Evans, and Satish Puri. GPGPU-accelerated Interesting Interval Discovery and other Computations on GeoSpatial Datasets–A Summary of Results. In *Proc. 2nd ACM SIGSPATIAL International Workshop on Analytics for Big Geospatial Data (BigSpatial-2013)*, Orlando, FL, Nov. 5, 2013. ACM.